

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-102761

(43)Date of publication of application : 16.04.1996

(51)Int.Cl.

H04L 12/64
H04L 12/28
H04L 29/06

(21)Application number : 06-261280

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 30.09.1994

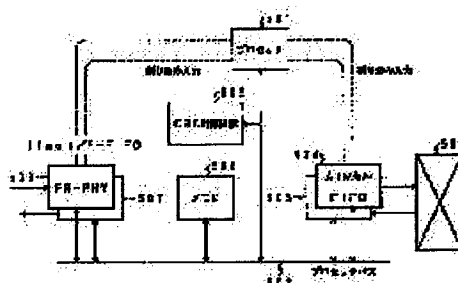
(72)Inventor : HASHIMOTO MIKIO
TSUNODA KEIJI
SHOHATA YASURO

(54) PROTOCOL CONVERTER AND PROTOCOL CONVERSION METHOD

(57)Abstract:

PURPOSE: To avoid waste and to accelerate an error detection code calculation processing speed by performing an error detection code calculation processing to be performed before and after protocol conversion only once to an overlapping part in transmission information.

CONSTITUTION: Cells inputted from an ATM switch 509 are stored in a buffer 505, then read in a processor 501, reconstituted into packets in a memory 503, ATM-FR converted and transmitted through an FR-PHY 507 to a line. On a reception side, when layer 2 packets are detected in the FR-PHY 506, data are transmitted to the memory 503 in the FR-ATM conversion process of the processor 501 and the packets are converted into an AAL 5 form, divided into the cells, written in a cell buffer 504 for each connection and outputted to the ATM switch 509. In an error detection processing at the time of the protocol conversion in transmission and reception, for a part whose contents after the conversion do not change, a processing result on a transmission side is utilized and the error detection processing is omitted on the reception side.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.*** shows the word which can not be translated.

3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In the protocol conversion equipment which carries out protocol conversion of the 1st transmission information to the 2nd transmission information according to the 2nd protocol the 1st transmission information according to the 1st protocol -- inputting -- this -- The 1st error detection processing means which processes the error detection processing in this 1st protocol within at least one processor to the 1st transmission information according to said 1st inputted protocol, As opposed to the part from which the contents change after this protocol conversion among said 2nd transmission information by which protocol conversion was carried out to said 2nd protocol As opposed to the part from which the contents do not change after this protocol conversion while processing the error detection processing in this 2nd protocol within at least one processor Protocol conversion equipment characterized by providing the 2nd error detection processing means using the processing result by said 1st error detection processing means.

[Claim 2] Between the 1st transmission information according to an ATM protocol, and the 2nd transmission information according to protocols other than an ATM protocol In the protocol conversion equipment which processes the processing of at least a part which includes error detection processing among these protocol conversion processings using at least one processor while performing protocol conversion processing to this transmission information A means to perform error detection processing in the protocol before protocol conversion to said 1st and 2nd transmission information, Protocol conversion equipment characterized by providing a means to perform error detection processing in the protocol after conversion, to the part from which the contents changed in protocol conversion among said 1st and 2nd transmission information.

[Claim 3] Between the 1st transmission information according to an ATM protocol, and the 2nd transmission information according to protocols other than an ATM protocol this, while performing protocol conversion processing mutually to the 1st and 2nd transmission information In the protocol conversion equipment which performs processing of at least a part which includes error detection processing among these protocol conversion processings using at least one processor Protocol conversion equipment characterized by providing a means to perform error detection processing of this duplication part only at once before and after conversion when a duplication part exists in said transmission information before and behind protocol conversion.

[Claim 4] Between the 1st transmission information according to an ATM protocol, and the 2nd transmission information according to protocols other than an ATM protocol While performing predetermined error detection processing based on the step which inputs said transmission information according to said one of protocols in the protocol conversion art which performs protocol conversion processing to this transmission information, and said transmission information The protocol conversion approach characterized by having the step which performs

format conversion of this transmission information, and the step which performs error detection processing in the protocol after conversion to the part from which the contents changed with format conversion among said transmission information.

[Translation done.]

*** NOTICES ***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.*** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention relates to the equipment and the approach of interconnecting the communication link of interface formats other than ATM and ATM, and relates to the protocol conversion equipment and the approach of changing mutually the data transmitted in the form of each interface proper.

[0002]

[Description of the Prior Art] The equipment formed in order to transmit information mutually between two communication modes is protocol conversion equipment (referred to as Inter Working Function.). Here, the technique of the conventional IWF which has an ATM interface by making into an example the protocol conversion equipment (hereafter referred to as IWF) to which between ATM and Frame Relais is connected is explained.

[0003] A Frame Relay is a leading protocol when accelerating the existing packet network represented by ITU-T recommendation X.25. Transmission speed and the delay engine performance are improving in the Frame Relay (it abbreviates to FR hereafter), having a user interface near the existing packet communication, as a result of simplifying protocols, such as resending control which was being performed for every link, with the improvement in the transmission quality of a channel.

[0004] On the other hand, ATM is a protocol which unifies various communication links, such as voice and data. In ATM, routing of the information is stored and carried out to a comparatively short packet called a cel.

[0005] The conceptual diagram of IWF at the time of being mounted in drawing 11 in a swap device is shown. This IWF has two kinds of interfaces, ATM and FR.

[0006] The cel inputted from the ATM interfaces 11121-11123 is switched with the ATM switch 11101, and the cel of the connection by whom it was beforehand set of them is inputted into IWF 11111-11113. IWF changes into FR format the packet inputted in the form of a cel, in addition performs required processing, and outputs it to the Rhine interface of FR. IWF decomposes into a cel the packet inputted from the interface of FR, and also performs processing sent out to ATM.

[0007] About IWF functional specifications, it is ITU-T recommendation I.365.1. Specific Convergence Sublayer(FR-SSCS) between Frame Relaying and It is set to B-ISDN. On a specification, by connecting each protocol function (Q. 922core, FR-SSCS (Fame Relaying Service Specific Convergence Sublayer)) simply, IWF is easily realizable, as shown in drawing 12.

[0008] By the way, two protocol functions have been conventionally mounted by the respectively following approaches.

[0009] (Example of an assignment configuration of hard/software) Drawing 13 is the conceptual diagram of the functional arrangement inside the conventional IWF. In drawing 13, 11301, 11302, and 11303 express a processor, memory, and a processor bus, respectively. 11311, 11321, and 11322 are the FR-PHY section constituted by hardware, respectively, the AAL5-CPCS section, and ATM layer ability. Moreover, 11332 and 11332 are the Frame Relay protocol processings (it abbreviates to Q.922 processing below) and FR-SSCS processings which are performed by the processor, respectively.

[0010] The function which belongs to a part of layer 1 and layer 2 among the processing facilities of a Frame Relay protocol is realized as exclusive hardware 11311 which constituted layer 1 physical-layer processing in which ITU-T recommendation I.431 was followed, by the logic gate. Moreover, layer 2 processing in which Q.ITU-T recommendation 922 core is followed is mounted as software processing 11331 by the processor.

[0011] Moreover, ATM layer ability and the function of the AAL5-CPCS section are mounted as exclusive hardware among the functions by the side of ATM. And the service proper section of AAL and FR-SSCR are mounted as software processing by the processor.

[0012] From ATM layer ability to a CPCS function is mounted by exclusive hardware for a rate required of software processing not coming out, when it corresponds to a high-speed ATM interface.

[0013] (Outline of a processing flow) Next, the contents of the processing performed by IWF based on drawing 14 are explained more to a detail. In drawing 14, 11411, 11412, 11413, 11414, 11415, and 11416 are functional groups which process the input from a Frame Relay interface, and express a physical layer function part, Q.922 function part, AAL5 function part, an ATM layer function part, a shaping (empty cel insertion) function part, and an output cel buffer, respectively. Moreover, 11421, 11422, 11423, 11424, 11425, and 11426 are the functions to process the input from the ATM switch 11441, and express an input cel buffer, an ATM layer multiplexing function part, an empty cel / invalid cel processing facility section, AAL5 function part, Q.922 function part, and a physical layer function part, respectively. 11431 expresses an OAM function part common to I/O.

[0014] (Conversion to FR from ATM) The ATM cel inputted from the ATM switch 11441 is separated for every connection by the ATM layer function part 11422 based on the specification described by ITU-T recommendation I.361. In advance of separation of a cel, routing in the ATM switch 11441 and the routing tag information added for OAM are deleted.

[0015] An OAM cel is sent to the OAM function part 11431, and an empty cel / invalid cel is discarded by the function to treat each cel. The cel with the header value corresponding to the AAL connection set beforehand is passed to the AAL function part 11424, and is processed by the AAL function part 11424 concerned. In the SAR sublayer of the AAL function part 11424 which received the cel, referring to cel header information based on the specification described by ITU-T recommendation I.363, two or more cel payloads are connected and SAR-SDU is reconfigured. And processing of error detection etc. is performed by the CPCS sublayer function part of AAL. The method of performing these processings in hardware is learned widely, and is put in practical use. The ATM layer function part 11322 and the AAL5-CPCS function part 11321 which were shown in the block diagram of drawing 13 correspond to this.

[0016] The next processing is performed by the processor 11301 in the example of a configuration of drawing 13. CPCS-SDU stored in the buffer memory of a CPCS sublayer is read by the processor, and is transmitted to the buffer memory (not shown) of a processor. Among AAL functions, since it is only rewriting a part of field of the header of CPCS-SDU received from the CPCS sublayer, FR-SSCS is easily realizable by operating the data on buffer memory. And processing of Q.922 function part 11425 is shortly performed to this packet data. Processings here are multiplexing by the logical channel, and a replacement of DLCI. The DLCI field of the packet data passed from FR-SSCS is rewritten to the rewriting DLCI value which was able to be beforehand defined for every value of the.

[0017] The following processings are performed by FR-PHY11311 in the example of a

configuration of drawing 13 . First, about layer 2 packet, CRC is calculated based on the computational procedure of CRC-16, and it is added to the end of a packet. And a flag pattern is added before and after a frame. In order to prevent a flag pattern and the same pattern appearing by these contents of data, the bit data of a value 0 are inserted per 1 bit, original physical-layer (layer 1) processing, i.e., processing of line coding, is performed to this data, and data are sent out as line-out of a Frame Relay interface.

[0018] (Conversion to ATM from FR) The procedure almost same as the conversion to ATM from FR is followed. While differing somewhat multiplexes other connections and the cel of OAM suitably so that the transmitting rate as which the transmitting connection was beforehand determined in the ATM layer may not be exceeded, or when it is inadequate, it is that there is a shaping function part 11415 which presses down a transmitting connection's transmitting rate below to default value by inserting an empty cel compulsorily.

[0019] Thus, it is possible by combining the exclusive hardware of a certain ATM and Frame Relay from the former to realize IWF.

[0020] By the way, when these processings are mounted, exclusive hardware is not necessarily indispensable. If the approach of the communications protocol processing by the general-purpose processor is used, the rate to which processing by the processor of an ATM layer, AAL, and FR physics layer is equal to practical use may be able to attain. It not only reduces the cost which develops exclusive hardware, but by realizing the function conventionally realized by exclusive hardware by the software function of a high performance general-purpose processor, the cost cut by fertilization of a general-purpose processor part is expectable.

[0021] As already explained, it is easily realizable by combining the conventional technique simply to realize ATM-Frame Relay IWF by software processing itself. But this mounting approach has left the room of the optimization doubled with the property of processor processing, and can plan improvement in the engine performance, or reduction of cost by taking mode of processing suitable for processor processing. Moreover, even if the service based on another ATM layer ability or an AAL function will appear in the future by realizing ATM layer ability / AAL function by software processing, the hardware of the existing IWF remains as it is, and correspondence becomes possible.

[0022] However, improvement in the speed of the communication facility by FUTOWEA in ATM-FR interworking was difficult. This originates in the trouble of a processor processing proper one.

[0023] As a problem of a processor processing proper, the overhead of the transfer between processor-memory is mentioned first. The improvement in a rate of a high performance processor in recent years is the so-called von from which memory access becomes the limiting factor of processing speed in order that the improvement in a rate of surrounding storage may not catch up with a processor by one side, although it is remarkable. neumann The phenomenon of bottleneck became remarkable and has arisen. It is pointed out with the communication device which must especially process the new data which arrive one after another that this problem appears notably (D. Banks, Mprudence. "A High Performance Network Architecture for aPA-RISC Workstation". IEEE Journal on Selected Areas in Communication 1993 vol.10 No.1). Long duration use of the processor will be carried out for the mere copy from a part to another part with a primary storage instead of actuation of complicated data. Especially the thing for which a copy of data is performed for every hierarchy of a protocol when a hierarchical protocol is mounted as it was is waste of serious processor capacity and processor bus transfer capability.

[0024] It is also possible to perform data transfer in parallel to actuation of a processor by the DMA transfer as a certain solution from the former. But, in order to share a processor and storage and to perform parallel processing, the device of complicated contention control is required. Moreover, in a complicated device, an overhead occurs in itself and improvement only in the engine performance which carried out parallel processing of the copy cannot be expected in many cases. Furthermore, a complicated device will cause the increase of cost. In a high performance general-purpose processor in recent years, there are some which demonstrate the engine performance equivalent to a DMA transfer in the transfer between memory-memory by the parallel execution of the instruction actuation called a pipeline. Of course, while performing

data transfer using the processor, data processing by the processor cannot be performed, but this is the trade-off relation between taking in DMA equipment, and it is practically equal, also when the rate of data transfer uses a processor in a current technique, and also when using DMA equipment. If own data transfer capacity of a processor is used effectively, a bus master can perform comparatively high-speed data transfer processing by the simple hardware configuration of only a processor. Furthermore, it is necessary to reduce the unnecessary data transfer which suited the conventional protocol processing.

[0025] Apart from the problem of data transfer, there is unsuitable processing among the processor processings. For example, they are scramble/descrambling processing which repeats many simple bit manipulation, and the computation of CRC. Although a processor can process TETA of WORD width of face efficiently, it is because the effectiveness of processing of the data over WORD or the repeat processing to the bit field in WORD is not good.

[0026]

[Problem(s) to be Solved by the Invention] By conventional protocol conversion equipment and a conventional approach, computation of error detecting code was not able to be efficiently performed at a high speed. Moreover, since count of error detecting code had to be calculated about each of two packet formats, protocol conversion before and the conversion back, futility has arisen about count of the part which after conversion does not change, and improvement in processing speed had been barred.

[0027] This invention is made in view of the above-mentioned situation, and aims at offering the protocol conversion equipment and the approach which accelerated error detecting code computation.

[0028]

[Means for Solving the Problem] In the protocol conversion equipment which carries out protocol conversion of the 1st transmission information to the 2nd transmission information according to the 2nd protocol the 1st transmission information that this invention follows the 1st protocol -- inputting -- this -- The 1st error detection processing means which processes the error detection processing in this 1st protocol within at least one processor to the 1st transmission information according to said 1st inputted protocol, As opposed to the part from which the contents change after this protocol conversion among said 2nd transmission information by which protocol conversion was carried out to said 2nd protocol As opposed to the part from which the contents do not change after this protocol conversion while processing the error detection processing in this 2nd protocol within at least one processor It is characterized by providing the 2nd error detection processing means using the processing result by said 1st error detection processing means.

[0029] Moreover, this invention is between the 1st transmission information according to an ATM protocol, and the 2nd transmission information according to protocols other than an ATM protocol. In the protocol conversion equipment which processes the processing of at least a part which includes error detection processing among these protocol conversion processings using at least one processor while performing protocol conversion processing to this transmission information A means to perform error detection processing in the protocol before protocol conversion to said 1st and 2nd transmission information, It is characterized by providing a means to perform error detection processing in the protocol after conversion, to the part from which the contents changed in protocol conversion among said 1st and 2nd transmission information.

[0030] Moreover, this invention is between the 1st transmission information according to an ATM protocol, and the 2nd transmission information according to protocols other than an ATM protocol. this, while performing protocol conversion processing mutually to the 1st and 2nd transmission information In the protocol conversion equipment which performs processing of at least a part which includes error detection processing among these protocol conversion processings using at least one processor When a duplication part exists in said transmission information before and behind protocol conversion, it is characterized by providing a means to perform error detection processing of this duplication part only at once before and after conversion.

[0031] Moreover, this invention is between the 1st transmission information according to an ATM

protocol, and the 2nd transmission information according to protocols other than an ATM protocol. While performing predetermined error detection processing based on the step which inputs said transmission information according to said one of protocols in the protocol conversion art which performs protocol conversion processing to this transmission information, and said transmission information It is characterized by having the step which performs format conversion of this transmission information, and the step which performs error detection processing in the protocol after conversion to the part from which the contents changed with format conversion among said transmission information.

[0032]

[Function] In this invention, when a duplication part exists in the transmission information before and behind protocol conversion, error detecting code computation (for example, CRC computation) performed before and after protocol conversion is performed only at once before and after conversion to this duplication part.

[0033] For example, in the protocol conversion between ATM and a Frame Relay, a CRC sign is added to both of the packets of a format for error detection. Since the fields which are rewritten in process of protocol conversion, or are added / deleted are very few, as for two kinds of this packet, most data areas serve as a duplication part.

[0034] Since error detecting code computation is performed only at once before and after conversion to the duplication part which closes most data areas by this invention to having performed error detecting code computation twice into this common data part, i.e., a duplication part, since the function to process two kinds of interfaces was combined simply conventionally, improvement in the speed of protocol conversion processing is expectable.

[0035] Moreover, when performing error detecting code computation by processor processing, reduction of processing loads can be aimed at by [which is a processing load] coming size and substituting error detecting code computation only for once to this duplication part.

[0036]

[Example] Hereafter, the example of this invention is explained, referring to a drawing.

[0037] The basic configuration and actuation of one example of this invention are explained at the beginning of <the basic configuration of an example>.

[0038] This example explains the basic configuration in the case of applying processor processing to protocol conversion equipment (IWF equipment). Hereafter, this example is explained based on drawing 1 -6.

[0039] In drawing 1 501 a processor bus and 503 for a processor and 502 Processor memory, An output cel buffer for 504 to output an ATM cel to an ATM switch, An input cel buffer for 505 to input an ATM cel from an ATM switch, The Frame Relay physics layer processing facility which 506 receives Frame Relay line-in and generates layer 2 packet (it considers as FR-PHY below), The Frame Relay physics layer processing facility which 507 inputs layer 2 packet and outputs to a Frame Relay line (it considers as FR-PHY below), The CRC count function to calculate the CRC at the same time, as for 508, a processor writes data in memory, and 509 are ATM switches.

[0040] In drawing 2 , 601 finishes setting up an ATM cel in a receiving-side ATM layer ability process, finish setting up 602 to a packet, and change into a Frame Relay format. The ATM-Q.922 conversion functional process of processing a part of AAL5 (FR-SSCS) and Q.922 protocol (it is called below an ATM-FR conversion functional process), The Q.922-ATM conversion functional process of processing the part and AAL5 (FR-SSCS) of Q.922 protocol which 603 changes a Frame Relay packet and is decomposed into an ATM cel (it is called below a FR-ATM conversion functional process), 604 -- for an input cel buffer and 505, as for output FR-PHY and 506, an output cel buffer and 507 are [a transmitting-side ATM layer ability process and 605 / an OAM functional process and 504 / input FR-PHY and 509] ATM switches.

[0041] Drawing 3 shows the relation of the I/O device of a processor, each functional process, and data in ATM-FR conversion.

[0042] In drawing 3 an input cel buffer and 601 505 An ATM layer process, The empty cel processing facility subroutine which a cel header-function conversion table and 721 are called

for 712 from an ATM layer process, The OAM cel processing facility subroutine which 722 is called from an ATM layer process, The AAL (SAR) functional subroutine which 723 is called from an ATM layer process, 605 an OAM cel buffer and 741 for an OAM process and 732 An ATM-FR conversion functional process, The buffer 1,507 which the buffer 1,753 which the buffer 1,752 with which an AAL buffer and 508 have in a CRC count function, and 742 has 751 in the AAL buffer 741 has in the AAL buffer 741 has in the AAL buffer 741 is output FR-PHY.

[0043] Drawing 4 shows the relation of the I/O device of a processor, each functional process, and data in FR-ATM conversion.

[0044] In drawing 4 an I/O cel buffer and 601 504 An output ATM layer process, The OAM cel processing facility subroutine which a connection-function conversion table and 821 are called for 812 from an ATM layer process, The output cel display which shows that, as for 822, OAM output data exist to an ATM layer process, 823 is called from an ATM layer process. A **** AAL (SAR) functional subroutine, The output cel display which shows that, as for 824, AAL output data exist to an ATM layer process, 605 an OAM cel buffer and 841 for an OAM process and 832 A FR-ATM conversion functional process, The buffer 3,506 which the buffer 2,853 which the buffer 1,852 with which an AAL buffer and 508 have in a CRC count function, and 842 has 851 in the AAL buffer 841 has in the AAL buffer 841 has in the AAL buffer 841 is input FR-PHY.

[0045] Drawing 5 shows the format of each packet processed by IWF.

[0046] In drawing 5, 901 expresses the signal of the format transmitted in a Frame Relay transmission-line top. The information field where the header and information field where a flag pattern and 911 constitute two layers and, as for 912-914, 902,903 constitutes two layers, respectively, FCS, and 921 constitute CPCS-PDU, and 922-923 constitute CPCS-PDU, respectively, a trailer, PAD from which 924-928 constitute a trailer, respectively, CPCS-UU, CPI, Length, CRC, and 931-1,931-2 - 931-n are cels.

[0047] The configuration of this example at the time of using 2portRAM for drawing 6 is shown.

[0048] 1001 -- a processor and 1002 -- a processor bus and 1003 -- receiving memory and 1004 -- for an input cel buffer and 1007, as for transmitting-side FR-PHY and 1009, receiving-side FR-PHY and 1008 are [transmitting memory and 1005 / an output cel buffer and 1006 / a CRC count function and 1010] ATM switches.

[0049] Hereafter, this example is further explained to a detail.

[0050] In this example, a phrase called memory shall show not the main memory that is the usual processor memory but the "cache memory" in a current high speed processor. Although it is well known according to concomitant use of high-speed cache memory and comparatively low speed main memory that improvement in the speed and large capacity-ization of memory can be performed, in communications protocol processing, the conditions, i.e., the locality of access, on which a cache device operates efficiently are not satisfied about data at least. Moreover, since it has a timer in the interior in communications protocol processing, the indeterminacy of the execution time by performing swapping out of an instruction is a factor which makes mounting difficult. Therefore, it considers fastening an instruction/data to a cache and performing them here. Moreover, the processor peripheral device is also premised on connecting with not the so-called processor bus but the bus of cache memory, or an equivalent high speed bus. This shall build in FIFO so that the pipeline actuation in a processor may not be disturbed at the time of memory access.

[0051] Next, basic actuation of the ATM-FR-IWF equipment by the processor processing which will be the requisite for this example is explained in order of ATM-FR conversion and FR-ATM conversion.

[0052] Based on [outline actuation] drawing 1 and drawing 2, the outline of actuation of the process on the hardware used as the base of this invention and a processor is explained.

[0053] First, the cel inputted from the ATM switch 509 is accumulated in the input cel buffer 505. These contents are read by the processor 501 and treated in the process on a processor 501 henceforth.

[0054] The ATM layer process 601 starts the processing beforehand defined based on the value of a cel header. The cel of the connection corresponding to a Frame Relay is transmitted to the

processor memory 503, and the packet divided into the cel is reconfigured. The reconfigured packet data are passed to the ATM-FR translation process 602, and processing required for conversion of ATM and a Frame Relay is performed. The processing on a processor 501 finishes even here, and a packet is handed over by FR-PHY507 and transmitted to the Rhine interface.

[0055] In a receiving side, if layer 2 packet is detected by FR-PHY506, the FR-ATM translation process 603 on a processor 501 is started, data will be transmitted to the processor memory 503 and a packet will be changed into AAL5 (FR-SSCS) format. This packet is divided into a cel in the ATM layer process 604, and as it does not exceed the cel rate beforehand defined for every connection, it is written in the output cel buffer 504 as a cel style by the shaping function which is one function of the ATM layer process 604, and it is outputted to the ATM switch 509.

[0056] Next, based on drawing 3 and drawing 4, actuation is explained to a detail including the DS on processor memory.

[0057] [Conversion to FR from ATM]

(Starting timing of the input ATM process of a cel) The cel outputted from the ATM switch is stored in FIFO buffer 505. An interrupt signal will be inputted into a processor if the amount of the cel accumulated in FIFO buffer 505 exceeds the constant value defined beforehand. By this interrupt signal, the receiving-side ATM layer process 711 is started. Although notifying arrival of a cel to a processor is not based on an interrupt signal, for example, it is possible also by the periodical polling from a processor, since a high speed and useless monitor processing are [the response] generally more unnecessary than polling, and there are few processing loads of a processor, the interruption processing of interruption processing is more advantageous.

[0058] (Alignment of a cel input cutting tool unit) In case a cel is stored in a buffer, in order to make treatment of a processor easy, it is desirable to take alignment by the word unit. For example, in the bus interface with which 1 word consists of 8 bytes, 3 bytes of addition part (PAD) is prepared in the head of a cel, and the approach of storing so that it may always be similarly located while the header of one cel is WORD is learned. Generally, in treating a cel with a switch, it adds and treats several bytes of routing tag in 53 bytes of cel in many cases. By adding 11 bytes of routing tag, it is also possible to take alignment, using cel length as 64 bytes.

[0059] (A header is recognized by the cel head signal) The output of the cel head signal from an ATM switch is held in an input cel buffer, and it notifies to a processor side that it is the WORD cel head concerned again at the time of the WORD readout by the processor.

[0060] (The repeat and ATM layer process killing of a function corresponding to retrieval/of a table) The receiving-side ATM layer process 711 reads in order the data stored in the input cel buffer 505 in the condition of having aligned in this way. A processor searches the cel header value-function conversion table 712 set up beforehand, and performs the function described by cel header value correspondence of the above-mentioned table 712 at the same time it reads a cel header. Since a header's being effective is check ending in the physical layer of a receiving side before a cel passes an ATM switch, it is not necessary to perform the check of a header by force here. However, a header and a routing tag may be inspected for a switch functional test etc. The ATM layer process 711 performs only the count which was able to define beforehand the actuation which reads a repeat cel header and performs the function to correspond, and ends activation of a process. And if a cel is inputted into the input cel buffer 505 from a switch and the cel more than a certain defined number is accumulated, the ATM layer process 711 will be started again.

[0061] (Processing of an empty cel) Now, if a cel header is value VPI=VCI=0 which shows an empty cel, actuation which supports to FIFO abandonment of the data of the payload part (48 bytes) following the actuation, i.e., the header, described by the field where the cel header value-function conversion table 712 corresponds will be performed, for example. Abandonment of data is eliminating the contents of the head of the input cel buffer which should be read to a degree, without sending out data to the bus of a processor. By discarding data, unnecessary data can reduce the futility which occupies a bus. In the layer more than an ATM layer, since the payload of an empty cel is not referred to, it is satisfactory to eliminate a payload at all. the header which the processor read about the empty cel -- discarding .

[0062] (Notation of the function corresponding to a connection) The approach of describing the

function corresponding to a connection on a table has the approach of describing the start address of the subroutine which performs for example, an empty cel abandonment function. In this case, the VPI/VCI value of a receiving cel = the empty cel function 721 is performed by moving control to the address of the field corresponding to the entry of the table corresponding to 0.

[0063] (Treatment and the diagnosis of an invalid cel) About the invalid cel which an OAM function does not treat, you may discard in the user cel by which a connection setup is not carried out, and an OAM cel, without reading a payload like an empty cel. For example, the cel of the VPI/VCI value by which a connection setup is not carried out is discarded, without reading a payload. The cel of the header value which shows the meta-signaling channel in the interface device which does not treat a meta-signaling channel etc. is discarded similarly.

[0064] On the table, only the cel header value of the set-up connection and the type which an OAM function treats is described, and what is not contained in them is treated as an invalid cel. Although an empty cel may also exclude what is described on a table and may treat it, since the treatment of an empty cel and an invalid cel differs about cel statistical information, when the statistical information of a cel unit is required, suppose that it is another. [as well as an invalid cel]

[0065] (Search method of a table) In order for a cel to judge an OAM cel or a user cel, it is necessary to see the field of both a VPI/VCI value and PTI. In such a case, in order to inspect only each field, it is common to judge combining those decision that performed the AND operation in the specific bit of a header, and went to two or more fields.

[0066] By this approach, the number of steps required for header inspection and the amount of tables are reducible compared with the way of comparing the value of a cel header simply.

[0067] (Dynamic modification of a search method) Now, the technique of table retrieval improvement in the speed of a hash, a watch soldier, etc. who were known well is applicable to retrieval of a table.

[0068] In addition, in order to accelerate table retrieval, there is the technique of changing the array of a retrieval table according to the occurrence frequency of an input cel. But, conventionally, modification of header funiculus sequence was restricted per connection, and it was only changing the array of the table on which the connection's was described.

[0069] The retrieval time of a header table is dependent also on the number of setting connections of a table, or the occurrence frequency of the cel for every connection classification as well as a retrieval procedure. Moreover, although the retrieval procedure used as the retrieval time shortest also changes with these parameters, optimization of table retrieval time can be attained by changing a retrieval procedure dynamically.

[0070] However, by the conventional approach, even if it can optimize retrieval effectiveness about a connection's cel occurrence frequency, retrieval effectiveness cannot be optimized to the frequency change between a user cel and an OAM cel. Because, actuation of seeing the specific bit field for judging an OAM cel is because it was written in in the code of a program outside the connection setting table.

[0071] Here, it also makes it possible to change inspection routine dynamically by not fixing header inspection routine but using the header table for comparing a cel header as the rewritable table which described the retrieval procedure. The table which described the retrieval procedure should just specifically describe the address of the entry of the subroutine which performs the retrieval procedure in which the value of the connection identifier of a header and the value of a specific bit field are inspected, like description of the function of a connection unit.

[0072] Activation of retrieval is performed by repeating in order until it returns the return value of the retrieval termination as which activation of subroutines, such as the entry described by the table, for example, user connection distribution, activation of a correspondence function and separation of the OAM cel which carries out termination locally, and a turnover to an OAM process, was beforehand determined to one of subroutines.

[0073] And the OAM function performed in [the ATM layer ability of IWF] asynchronous rewrites the table which described the sequence of retrieval that retrieval time becomes the shortest based on the arrival frequency table for every header value of a receiving cel for every

fixed time amount.

[0074] For example, at the time of usual, the header should be first searched about the VPI/VCI value and then separation of an OAM cel and a user cel should be performed based on PT. It is more efficient to carry out actuation which carries out cel separation for every connection with a VPI/VCI value first, when there are many user cels. On the other hand, an F5 (VCI level) segment OAM cel can be distinguished only by PTI (PTI=100). When the rate of an F5 level OAM cel increases according to some causes, such as failure, the effectiveness of retrieval of the direction which separates an OAM cel by PTI first is good. Moreover, since a layer management entity common about all connections performs termination processing of an OAM cel in many cases, it may be necessary to dissociate and process an OAM cel for every connection.

[0075] On the contrary, when processing changes with cel headers like COM/EOM by the AAL sublayer, the functional subroutine corresponding to each header value may be performed by header functional distribution of an ATM layer.

[0076] Also while the OAM process is rewriting the above-mentioned table, header funiculus processing of an ATM layer may be started at any time. Normal retrieval of ATM layer processing searching the same entry twice, for example, if rewriting refers to a non-completed table etc. cannot be performed. This problem is solvable for business by having a table 2 sets and rewriting one by directing that ATM layer processing refers to the table which rewrote one more after rewriting was completely completed as an object for reference of an ATM layer.

[0077] Moreover, this rewriting actuation may not necessarily be aimed only at the retrieval time shortest, but may be changed with directions by the internal state of IWF, or decision of a high order control unit.

[0078] By for example, IWF which performs the retrieval subroutine which searches a table with normal operation about a VPI/VCI value first, and is distributed, and then is carrying out required processing with reference to PT/CLP. When the processor of IWF lapses into a congestion condition, CLP (Cell Loss Priority) of a receiving cel is searched first. If the cel of CLP=1 is which connection's cel, it can reduce the load of the reception of the cel of CLP=1, and retrieval processing of a corresponding table, and can make IWF recover a congestion condition by taking the measure discarded immediately.

[0079] Thus, by changing a retrieval procedure dynamically, it is expectable that the effectiveness of retrieval and processing is optimized in the range larger than before.

[0080] (Reception of an OAM cel) Next, the reception of an OAM cel is explained. If the header value of this cel shows an OAM cel, the OAM cel processing subroutine 722 will be called. The OAM cel processing subroutine 722 is sharing the OAM process 731 and the OAM cel buffer 732, and has the pointer in which it reads with the current write-in address, and the address is shown.

[0081] An OAM cel processing subroutine and an OAM process are performed as a separate process. The contention control of a buffer is realizable by the following approaches among both, for example.

[0082] (Contention control of the OAM cel buffer by the ring buffer) The OAM cel buffer 732 is a ring buffer, and if an OAM cel arrives, it will be written in the buffer area which the contents of a header value and the payload write in and begins from the address, and will be increased by the write-in address pointer by 1 cel buffer. On the other hand, an OAM process increases the pointer of the readout address read now by the single address, after read-out of one cel is completed. Before any pointer reaches to the ending address of a buffer, it is returned to a start address. Since an OAM cel processing subroutine is written in before writing, a pointer reads and it inspects that it is not the same value as a pointer, access of both processes does not compete. Since it is overflow of a buffer when a write-in pointer reads and the same address as a pointer is pointed out, the cell data which was planning writing is discarded.

[0083] An OAM process performs processing of an OAM cel repeatedly until it finishes processing of the cel of all the cels accumulated in the buffer. However, it prevents making it higher than an OAM process, barring activation of an ATM layer process, and a receiving cel buffer overflowing the execution priority of an ATM layer process. Therefore, also while an OAM process performs processing, an ATM layer process starts, and a new cel may be written in an

OAM cel buffer.

[0084] (Starting conditions of an OAM process) An OAM process is dormant when an OAM process finishes processing of all cels (i.e., when a readout pointer writes in and it catches up with a pointer). An ATM layer process makes activation of the once dormant OAM process resume by rising an OAM process at every activation termination, since idle.

[0085] An OAM process is performed with the priority which ranks second to an ATM layer process.

[0086] (Approach using the message of pointer delivery) It is very good in the method which transmits and receives the pointer of the write buffer address and the processed buffer address as a message, and controls a buffer exclusively between [other than the buffer control method using the ring buffer and pointer which were explained so far] an ATM layer process (OAM cel processing subroutine) and an OAM process. Since it is the same as that of the AAL function described below, the detail in the case of exchanging a pointer as a message is omitted. In this case, reception of a message becomes the starting conditions of an OAM process.

[0087] (AAL function) Next, a user connection's AAL functional subroutine is explained.

[0088] Arrival of the cel of the AAL connection to whom a header value suits the value described by the header function conversion table 712 like an OAM cel or an empty cel performs the subroutine described by the corresponding field. Here, the cel which specifies the AAL subroutine 723 should arrive.

[0089] (Buffer control corresponding to an AAL connection) Through AAL5 (FR-SSCS) connection, processing changes with PTI values of an arrival cel header. The cel of a tail of the packet of AAL5 is ATM-UU=EOM, and the remaining cels consist of cels of ATM-UU=COM.

[0090] (Treatment buffer configuration of the cel of COM) First, actuation when the cel of ATM-UU=COM arrives is explained. The AAL functional subroutine 723 performed from the ATM function stores the cel payload in the buffer defined for AAL5. An AAL function has the continuous buffer area AAL buffer 742 which can store two or more data of the maximum packet size beforehand defined through the connection at least. In drawing 3, the buffer configuration which can store three packets is shown and buffer 1-3,751-753 can store the maximum packet size, respectively.

[0091] (Treatment buffer overflow of a COM cel) The AAL functional subroutine has a write-in pointer holding the current write-in address of a buffer, and it updates the pointer, writing in the address with which this pointer shows the cel payload read from the input cel buffer 505. When the buffer length crosses the last address of a buffer exceeding the maximum length set beforehand through a connection, the data stored by then are discarded as an invalid, and the variable of a write-in pointer and others and the value of the below-mentioned CRC count function are returned to the initial value defined beforehand. In AAL5, this maximum length is specified as Max SDU Deliver Length at the time of a connection setup. When AAL5 connection's EOM cel is discarded by congestion etc., such a phenomenon may occur.

[0092] (Count of CRC) The payload of a receiving cel is written in the address which said buffer write-in pointer points out. At this time, count of CRC32 is also performed to coincidence. For calculating CRC32 by software, it is $n \cdot n/2$ count of an exclusive OR (it changes with data) is needed by n shifts and average to the data of bit. This is quite large compared with the time amount of data transfer. For example, if the penalty of the transfer to FIFO and memory is assumed to be 4 times of the system clock of a processor, respectively, the 1 word (32 bits) transfer time will serve as eight clocks. But, in order to calculate 1-word CRC, the time amount of $32 \times 3/2$ clock is needed at least. By accelerating CRC count, processor processing of AAL5 function is accelerable beyond optimization of a memory transfer.

[0093] Here, using the auxiliary exclusive hardware CRC count function 508, the hardware incorporates the data to the writing and coincidence to memory by the processor, and performs CRC count to them. That for which the circuit of CRC count used the shift register is known widely.

[0094] By the initial state, the register value of a CRC count circuit is first set as 0. A count circuit performs the next count, when a value is written in the AAL buffer area 742 on the memory of a processor. Even if a processor accesses other memory areas, for example, the field

of a program or a stack, a CRC count function is disregarded. As [the] does not use the value of general CRC (X), i.e., $M(X) X^k \bmod G$, to which the error-checking code in AAL5 is called CRC32, and $M(X)$ is made into the example of information, and it makes $G(X)$ a generating polynomial. In CRC32, what took the one's complement of the sum of mod G ($M(X) X^{32} + (X^{31} + \dots + X + 1) X^k$) (X) and (mod 2) is used. This is equivalent to what added constant value to the result of CRC which makes general $G(X)$ a generating polynomial. Below, in order to simplify explanation, it explains based on the count approach of general CRC.

[0095]

[Equation 1]

$$\text{result} = (R X^N + D) \bmod G(X)$$

R	レジスタに保持されている値
D	メモリに書き込まれるデータ
N	バスの幅 (bit)
G (X)	生成多項式

When the write-in address output from a processor to memory is the range of the buffer defined beforehand, it is easy to constitute a circuit where a value is written in this CRC count circuit, and CRC count is performed. However, the width of face N of a data bus must be the integral multiple of the degree of generating-polynomial $G(X)$. The result of CRC count is stored in a CRC count circuit with transfer termination of the cel from which the degree of $G(X)$ constitutes a packet at the time of $N = 32$ since a cel is also 48 bytes (32 bits), i.e., 4 bytes, of multiple in 32. What is necessary is just to calculate CRC about the data which reversed the first 4 bytes (32 bits) of a packet head cel for count of CRC32. In addition, although the case where a data bus was 32 bits here was described, even if data bus width of face is set to 64 bits, this condition is still satisfied.

[0096] (rate of CRC count) If it operates at a high speed enough so that a CRC count circuit is completed in the data write cycle from a processor to memory, though a processor writes data in memory continuously, CRC count can be performed correctly.

[0097] On the other hand, when not ending while CRC is 1 cycle data writing, while the circuit which detects that CRC count is continuing is established in this circuit and count is continuing, it is possible by suspending writing for a waiting state to calculate right CRC.

[0098] (How to calculate processing CPCS-LP/CI at the time of COM/EOM) If there is a cel from which the bit of CLP or CI of the header of a receiving cel is 1, variable CPCS-LP or CPCS-CI which the AAL function concerned has locally will be set as 1. These variables correspond to primitive CPCS-LP of the same name of the CPCS sublayer described by ITU-T recommendation I.363.6, and CPCS-CI, and a value 0 is assigned to an initial state.

[0099] The above processing is performed, an AAL functional subroutine is ended, and activation returns to an ATM layer. A subroutine is performed at every arrival of the cel of the connection concerned, and a packet is accumulated in a buffer.

[0100] (Processing of EOM) The case of EOM to which ATM User Information of a receiving cel expresses the end of a packet is compared with the value from which the payload transfer was completed and the value of a count circuit was expected after the completion of count of CRC. Since a bit error or cel abandonment will have occurred if the result of a comparison is inharmonious, the packet concerned is an error (invalid) and initializes the variable of the AAL functional subroutine 723.

[0101] the time of EOM reception -- data -- effective/invalid -- also when it is any, the internal register of a data write-in pointer, the number of receiving cels, CPCS-LP, and each variable of CPCS-CI and a CRC count function is returned to the initial state defined beforehand.

[0102] (ATM layer process (AAL subroutine) -> data delivery to an ATM->FR process) When an effective packet is detected, an AAL subroutine notifies the start address of a buffer 751, a packet size, and the value of CPCS-LP/CI to the ATM-FR translation process 741. An ATM-FR

translation process is started by this notice. Specifically, as for this notice, the message box function of OS is used.

[0103] (EOM post process) Next, after the ATM-FR process took out the address of the buffer 752 which carried out processing termination from the message box and considers as the address of a write-in pointer, an AAL subroutine initializes other variables and ends a subroutine. Since buffer areas are insufficient if a message box is empty, the value which shows that the buffer addresses, such as a value 0, are invalid to a write-in pointer, for example is assigned, and storing of a cel which reaches a degree is stopped.

[0104] (Relation between a burst and the amount of buffers) When the input by the side of ATM occurs burstily and the output by the side of a Frame Relay does not meet the deadline, such a phenomenon arises. In order to permit a big burst, it is necessary to increase the number of the buffer contained in the AAL buffer 742. If the arrival frequency of a cel is always fixed, the number of a buffer requires only two pieces, the object for AAL assemblies, and the object for a Frame Relay output.

[0105] (Cure against buffer overflow) Whenever the cel of the AAL connection concerned arrives also in a condition with a write-in invalid pointer, an AAL subroutine is started, and data abandonment actuation is performed. If the address of the buffer with which the ATM-FR translation process ended processing is notified to the message box, the address will be written in, initial setting will be performed as a pointer, and storing of a cel will be started.

[0106] (ATM-FR translation process) Next, an ATM-FR translation process is explained.

[0107] (Format conversion) Now, an ATM-FR process processes data next. The ATM-FR translation process 741 is started by receiving the information about a packet as a message from the AAL functional subroutine 723.

[0108] (Separation of a control message) The packet reconfigured by the AAL buffer is the format of CPCS-PDU921 that the trailer was attached to the tail. Modification required for a part for this header unit 922 is added, it changes into the layer 2 (it considers as the following L2) packet format of FR, and an ATM-FR translation process transmits a part for a header unit 922, and information field 923 to FR-PHY.

[0109] A translation process recognizes the data link identifier first contained in a header. A data link identifier belongs to a receiving connection, it checks that packets are not control/maintenance message, and parameter setup processing is started. If packets are control/maintenance message, it will notify to an OAM process and predetermined processing will be performed.

[0110] (Rewriting of a header) Conversion in L2 packet format of FR only rewrites the value of some fields of the headers of a packet.

[0111] What took OR of the value of CPCS-LP, i.e., a receiving connection's CPCS-LP value, and the value of the data priority (Data Eligibility) included in a packet is written in the field of the data priority (Data Eligibility) of a packet.

[0112] What took OR of the own value of the FECN field of the original packet and the value of CPCS-CI in FECN is written in the field of FEC of a packet.

[0113] (DLCI rewriting) When there is data link identifier DLCI rewriting ***** for every physical link, based on the table which corresponded the appearance side DLCI the entering side DLCI, the appearance side DLCI is searched for the entering side DLCI as a key, and DLCI is rewritten.

[0114] (Data output to FR-PHY) Now, conversion of a packet format is ended. The header and information field of this packet are transmitted to hardware FR-PHY507 which performs transmitting processing of L2 packet.

[0115] Transmitting processings are specifically the flag pattern before and behind a frame, addition of FCS (CRC16), 0 insertion, and physical-layer processing (line coding).

[0116] (Problem of the transfer time) In this example, since the buffer for performing SAR processing of an ATM cel is shared with the buffer for data output to FR-PHY, while performing data transfer to FR-PHY, an ATM layer process cannot access this buffer. It is required to take a sufficiently large capacity of FIFO so that overflow of an input cel buffer may not arise between the data transfer to FR-PHY.

[0117] (Transfer procedure) First, the directions with which a processor starts the output of a new packet to FR-PHY are published, a processor reads the data which begin from the address which shows the head of the packet beforehand prepared for each AAL connection correspondence, and it writes in the register for data output prepared in said FR-PHY. Termination of all data writing publishes the data output termination directions containing the data length of a packet to FR-PHY.

[0118] This processing is very good in the mounting gestalt that direct transmitting initiation to the DMA function which accompanies FR-PHY, and read the contents of processor memory by the DMA function, and a processor transmits to data processing/transmitting function of FR-PHY, and processes.

[0119] However, as for the complicated function accompanied by bus mediation like DMA from a viewpoint of simplifying a hardware function, avoiding is desirable. In the communications protocol processing explained until now, in order that memory access may occupy big specific gravity, it is because processing cannot advance in the condition that DMA occupies the bus. However, when a processor is used about the bus interface of FR-PHY and interprocessor, in order that data transmission efficiency may influence direct-processing time amount, it is desirable to use a high-speed thing.

[0120] There is the approach of connecting port of one of the two to the DMA bus of FR-PHY considering the memory secured for the buffer for packets of a FR-SSCS function as RAM of 2port(s) as a means to avoid the problem on which a processor bus is occupied between data transfer, simplifying a bus mediation device. The block diagram is shown in drawing 6. If it does in this way, other processings can be advanced also during the data transfer of FR-PHY. However, 2portRAM is complicated as compared with single portRAM, and will cause cost quantity.

[0121] (Processing in FR-PHY) In FR-PHY, based on the computational procedure of CRC-16, CRC is calculated about the contents of the input data, and it is added to the end of a packet. And a flag pattern is added before and after a frame. In order to prevent a flag pattern and the same pattern appearing by these contents of data, the bit data of a value 0 are inserted by 1 bitwise, original physical-layer (layer 1) processing, i.e., processing of line coding, is performed to this data, and data are sent out as line-out of a Frame Relay interface.

[0122] (Termination of an ATM-FR translation process) After the ATM-FR translation process 741 finishes the data output to FR-PHY, the address of the buffer which finished the output is outputted to the message box of the AAL subroutine 723, and this buffer is again used for storing of a cel. Above, an ATM-FR translation process is ended.

[0123] (Conversion to ATM from FR) Next, it explains, referring to drawing 4 about the conversion function from FR to ATM.

[0124] (Physical layer) In receiving-side FR-PHY506, the processing based on ITU-T recommendation I.431, i.e., line-in, a clock extract, bit playback, descrambling, frame synchronization, and L two-frame detection (time fill removal) are performed to the signal inputted from Rhine of a Frame Relay interface. PHY-FR has the above buffer by two frames of maximum length permitted, and it continues the actuation which receives the data inputted from Rhine and is stored in a buffer after notifying arrival of a packet to a processor also until a transfer of this packet is completed. moreover, FR-PHY -- setting -- detection of L2 frame -- being concurrent -- this -- CRC count of L2 frame is also performed.

[0125] (Q. 922-frame detection) L2 Protocol Data Unit (henceforth, PDU) of a Frame Relay takes the frame structure shown in 911 of drawing 5. If the flag pattern by the side of frame termination is detected, FR-PHY will input an interrupt signal into a processor, and will notify arrival of a frame.

[0126] (Starting of a FR-ATM translation process) The processor which received interruption starts the FR-ATM process 841. As compared with the value from which the result to which FR-PHY506 calculated CRC about received data is expected, the receiving-side FR-ATM process 841 will direct abandonment of a packet to PHY-FR as an error, if both differ. This result is displayed on the register of FR-PHY which was mapped on the main memory of a processor and which was defined beforehand.

[0127] (Preparation of a header before a readout) For conversion to FR-ATM, a processor reads the data for 1 cel payload with which the header of L2PDU is contained from FR-PHY, and performs the following processings.

[0128] (Reservation of a packet storing buffer) The data length of this packet is displayed on the register of FR-PHY defined beforehand, and a processor gets to know the byte count of data which reads this and performs a transfer from FR-PHY. If the data length is in violation on protocol agreement, abandonment of this packet will be directed to FR-PHY. The buffer of the data transfer point has the continuous buffer area AAL buffer 842 which can store two or more data of the maximum packet size allowed through the connection concerned like the case of conversion to ATM-FR. In drawing 4, the buffer configuration which can store three packets is shown and buffer 1-3,851-3 can store the maximum packet size, respectively.

[0129] If only the magnitude which is enough to change not only the maximum length of Q.922 protocol but maximum length's packet into AAL5 format for the buffer length is taken, it will become unnecessary here, to perform migration fellow calm doubling of data in protocol conversion.

[0130] Moreover, in order to simplify the handling by the AAL subroutine, as for a buffer, it is desirable to consider as the die length of the integral multiple of a cel payload, i.e., the integral multiple of 48.

[0131] (Recognition of a header: Separation of a control message) The FR-ATM translation process 841 recognizes a data link identifier. A data link identifier belongs to a receiving connection, it checks that packets are not control/maintenance message, and parameter setup processing is started. If a data link identifier shows control/maintenance message, processing of the packet concerned will be passed to the OAM process 831, and a FR-ATM process will be ended.

[0132] (Rewriting of a header parameter) Most header parameters are used with a value as it is like the conversion to FR by the side of opposite from ATM. Only FECN, BECN (Forward/Backward Explicit Congestion Notification), and DLCI are changed.

[0133] (DLCI rewriting) When there is data link identifier DLCI rewriting ***** for every physical link, based on the table which corresponded the appearance side DLCI the entering side DLCI, the appearance side DLCI is searched for the entering side DLCI as a key, and DLCI is rewritten.

[0134] (Rewriting of FECN and BECN) The value of variable CPCS-CI to which the AAL function 623 received in the opposite side AAL reception function has a value 0 in the BECN field is assigned to the FECN field, respectively. Now, conversion of a packet is ended.

[0135] (FR-SSCS function-AD) Next, a packet is changed into the AAL5CPCS format 921, and a CPCS trailer is added. A trailer consists of a format as shown in 924. When dividing a packet per cel, it prevents storing a trailer ranging over between cels, and in order that a trailer may be located in the part with the always same last cel and may make recognition of a trailer easy, the variable-length field called PAD is added between a trailer and information field. A PAD field is filled up with OKUTEDDO of a value 0.

[0136] (FR-SSCS function: trailer) There is the field of CPCS-UU, CPI, Length, and CRC in a CPCS trailer body. CPCS-UU is set as a value 0 by FR-SSCS. Although the directions of CPI are still undecided, burying by the value 0 for the time being is specified. The Length field puts in the number of octets of a payload. The count result of CRC32 is stored in the CRC field. Now, the packet of an AAL5-CPCS format is able to be prepared. The start address and the number of cels of this packet are passed to the input queue of the AAL subroutine 823 corresponding to a FR-SSCS connection, and a FR-ATM translation process is ended. As for this notice, the message box function of OS etc. is specifically used.

[0137] (Data delivery to an AAL function) The output side AAL function 823 is called from the output side ATM layer process 601. In an initial state, first, an AAL function reads the start address and the number of cels of an output packet from the input queue from the FR-ATM translation process 841, and rewrites the output cel display 824 to those with a cel.

[0138] When a queue is empty, an output cel display is ended as having no cel.

[0139] When initialization is completed, the state variable with which the number of transfer cels

is expressed first is increased one time, as compared with the number of transfer cels passed from the FR-ATM translation process 841, if it is transfer termination, COM will be written in EOM, otherwise, PT and PT of the header data with which connection data were written in beforehand will be outputted to an ATM-SW buffer. And 48 bytes of data for 1 cel payload are read from the memory which a start address shows, and it writes in an ATM-SW buffer. After finishing a transfer, a start address is increased by 48 bytes and the value is held.

[0140] If it is transfer termination, two variables, a redirecting address and the number of transfer cels, will be returned to an initial state, the start address and the number of cels of an output packet will be read from the input queue from a FR-ATM translation process, and a variable will be initialized. When there are no data in a queue, an output cel display is ended as having no cel.

[0141] Also when a transfer of a packet is the middle, an AAL subroutine is ended on the completion of a transfer and the peach of one cell data, and control is returned to an ATM layer process.

[0142] (Transmitting-side ATM layer process) A transmitting-side ATM layer process assigns the connection of OAM or a user the right of an output of a cel in order according to the algorithm defined beforehand so that the traffic parameter which has agreed for every user connection at the time of a connection setup may be protected. The connection and traffic parameter which are set up are described by the header function conversion table 812, and an ATM layer process determines the schedule which outputs a cel based on the output cel display 822,824 of this table and each connection.

[0143] Each called connection cel output subroutine outputs one cel for every one-time call as explained previously.

[0144] When the connection who should output a cel in the case of scheduling cannot be assigned, an empty cel output is assigned. Although an empty cel output may have the subroutine of an empty cel output only, since an empty cel is a fixed pattern, the output function of an empty cel pattern is included in an ATM-SW buffer, and a processor may be made to direct an empty cel output.

[0145] (Priority of an ATM layer process) At the time of starting, an ATM layer process reads each output cel display corresponding to a connection once, and determines the schedule of a cel output. Therefore, the cel output subroutine of a connection without an output cel is not called. In the process in connection with a series of FR-ATM conversion, since the priority is the highest, an ATM layer process does not need to take into consideration that each connection's output cel condition changes during activation of an ATM layer process.

[0146] Although it is not concerned with the existence of an output cel, but the schedule is beforehand defined based on the table 812 and each connection's cel output subroutine may be called, there is futility to which the subroutine of a connection without an output cel is called repeatedly.

[0147] Thus, IWF which carries out the interconversion of the Frame Relay to ATM is realizable.

[0148] [Two or more interfaces] Although the AAL connection and the Frame Relay interface showed only one example by old explanation, IWF has two or more Frame Relay interfaces, and also when two or more ATM connections are connected to it, the hardware / software structure where it already explained can be applied. In order to correspond to two or more connections, it is easily realizable when only the number of connections increases the buffer 742,842 of AAL, and a connection's variables for buffer control (write-in pointer etc.) and FR-PHY506. About a CRC count function, it can respond to two or more connections by evacuating the internal state of the CRC count function 508 to processor memory for every transfer of 1 cel payload. Moreover, a count function may be shared, without preparing a CRC count function in every ATM-FR (drawing 3) and FR-ATM (drawing 4).

[0149] In this example, a packet is once reconfigured on the memory of a processor from the peripheral device of a processor, a packet format is changed, and it is outputting to the peripheral device. Reduction of processor bus pulse duty factors = a configuration which transmits immediate data to a peripheral device from a peripheral device can also be realized from a viewpoint of reduction of the count of a data transfer, without reconfiguring a packet on

processor memory.

[0150] In the above, the basic configuration of this example has been explained.

[0151] Below, the CRC count function of this example is explained to a detail.

[0152] <an example of the protocol conversion equipment between AAL5 and Ether> — the count abbreviation approach of CRC in the protocol conversion between AAL5 and Ether(s) with the same method of CRC is explained first.

[0153] Drawing 7 shows the packet format of ATM, AAL5, and Ethernet (IEEE802.3).

[0154] In drawing 7, in 1101, the header of a packet and 1115 express the information bureau of a packet, and, as for 1116, the packet of Ethernet and 1111 express FCS (CRC32) of a packet, respectively.

[0155] Moreover, the header of a packet consists of a transmitting first-move ground 1112, a sending agency address 1113, and die-length information 1114.

[0156] 1102 is CPCS-PDU of AAL5 and consists of a header 1121, an information bureau 1122, PAD1123, CPCS-UU1124, and each of CPI1125, length1126, and CRC32 (1127).

[0157] 1103-1 to 1103-n is the ATM cel generated or received by decomposition or assembly of CPCS-PDU.

[0158] Drawing 8 shows an example of the equipment configuration in the case of performing protocol conversion between AAL5 and Ether.

[0159] drawing 8 — setting — 501 — a processor and 502 — for an output cel buffer and 505, as for an Ethernet interface (Ether-I/F) and 508, an input cel buffer and 1206 are [a processor bus and 503 / buffer memory and 504 / a CRC count function and 508] ATM switches.

[0160] (Difference between a Frame Relay and Ethernet) In the basic configuration of this example mentioned above, although IWF which interconnects a Frame Relay and ATM was explained, IWF which interconnects ATM and Ethernet (IEEE802.3) similarly can be considered. This interface specification of IWF is ATM. forum LAN Emulation Over Although there is ATM etc., simply, SSCS (service dependence part) of AAL5 is NULL, and decides to put the packet of Ethernet there and to be crowded here.

[0161] What is necessary is just to also transpose interconnect of ATM and Ethernet to Ether-I/F1206 which performs transmission processing of Ethernet for FR-PHY of a Frame Relay like drawing 8. Also about an art, it can guess easily from the basic configuration of this example mentioned above with extent which reflects the difference in a packet format in header rewriting. Here, only the difference with the case of the above-mentioned Frame Relay is explained.

[0162] First, since QOS is not defined by Ethernet, it cannot necessarily surely output in time amount with a packet. While the collision of a frame has occurred, sending out of a frame cannot be performed. A frame must be discarded, if arrival of a new frame continues when sending out of a frame cannot be performed.

[0163] In order to solve this problem, there is a method of performing a flow control by the ATM connection side. Or you may use permitting frame abandonment for an ATM connection as a connection without QOS as a premise.

[0164] In a Frame Relay, the error detecting code of a packet is CRC16, and the count approach is different in CRC32 of AAL5. But, since the error detecting code of the packet of Ethernet is the same as CRC32 of AAL5, count of CRC32 of an invariable region to the format conversion of the frame of AAL5 and Ethernet is omissible. This invention is explained below.

[0165] Error detecting code FCS of the packet 1101 of Ethernet is expressed as follows.

[0166]

[Equation 2]

$$\begin{aligned}
 C(X) \bmod G(X) &= (A(X) X^{L_B} + B(X)) \bmod G(X) \\
 &= (A(X) X^{L_B}) \bmod G(X) + B(X) \bmod G(X) \\
 &\dots (1)
 \end{aligned}$$

$A(X)$ パケットのヘッダ部分1112, 1113, 1114の多項式表示
 $B(X)$ パケットの情報部分1115の多項式表示
 $C(X)$ パケット1101全体の多項式表示
 L_B パケットの情報部分1115のビット数
 $G(X)$ CRC32の生成多項式

On the other hand, the error detecting code of CPCS-PDU1102 is expressed as follows.

[0167]

[Equation 3]

$$\begin{aligned}
 V(X) \bmod G(X) &= (W(X) X^{L_T} + T(X)) \bmod G(X) \\
 &= ((A'(X) X^{L_B} + B(X)) X^{L_T} + T(X)) \bmod G(X) \\
 &= (A'(X) X^{(L_B + L_T)} + B(X) X^{L_T} + T(X)) \bmod G(X) \\
 &= \underbrace{(A'(X) X^{(L_B + L_T)} \bmod G(X))}_{\text{第1項}} + \underbrace{(B(X) X^{L_T} \bmod G(X))}_{\text{第2項}} + \underbrace{T(X) \bmod G(X)}_{\text{第3項}} \\
 &\dots (2)
 \end{aligned}$$

$V(X)$ CPCS-PDU1102の多項式表示
 $W(X)$ CPCS-PDUの情報部分の多項式表示
 $A'(X)$ パケット1102のヘッダ1121のフォーマット
 変更された多項式表示
 $B(X)$ パケット1102の情報部分1122の多項式表示
 $T(X)$ CPCS-PDUのCRCを除いたトレイラ部分
 1123, 1124, 1125, 1126 の多項式表示
 L_T パケットのトレイラ部分のビット数

(2) **** 2 term can deform as follows.

[0168]

[Equation 4]

$$\begin{aligned}
 (B(X) X^{L_T}) \bmod G(X) &= ((B(X) \bmod G(X)) X^{L_T}) \bmod G(X) \\
 &\dots (3)
 \end{aligned}$$

That is, in order to ask for (3) types, the 2nd term as a result of (1) type can be used.

[0169] Now, when CRC was calculated by the processor, that the operation of the number-of-bits order of an operand is the need touched with the basic configuration of this example mentioned above. Therefore, in order to obtain (3) from the 2nd term $(B(X) \bmod G(X))$ of the right-hand side of (1), they are $3/2 \times L_T$. What is necessary is just to perform the bit shift / XOR operation of a time. L_T It is the number of bits of a trailer, and it is very few if it compares with the number of bits of the whole packet, although $64 - 440$ operations are required.

[0170] When maximum length 1600octet of the packet of Ethernet and an average of 984 octet (s) of shortest length 368octet are compared with die-length 31octet (average of 55 and 8) of an average of a trailer part, the percentage of occupying to all the packets of a trailer part is about 3%.

[0171] Each processing time is considered. The penalty of the transfer to FIFO and memory is assumed to be 4 times of the system clock of a processor like the basic configuration of this

example mentioned above, respectively. The 1 word (32 bits) transfer time of a payload part serves as eight clocks, and the transfer times of a total payload are 7872 clocks with 984x8 clocks (reading + writing). CRC count of hard wafer processing is performed to this and coincidence.

[0172] By CRC count of a trailer part, 4 (word)x4 clocks (only reading), and a shift / XOR operation cut [memory access] the average length of trailer length, and CRC count of 31octet in a total of 388 clocks by $248 \times 3/2$.

[0173] The percentage of the processing time of the trailer section to the payload section is about 5% in $388 / 7872 = 0.049$.

[0174] Since alignment is not necessarily taken by 4 bytes, a trailer cannot necessarily use the CRC count circuit which operates by 4 bytes of word unit in this count.

[0175] In order to calculate the trailer section by exclusive hardware, the CRC count circuit which can set a unit as arbitration for the count of a octet unit, i.e., shift operation, 8 times is required. But, it is LT as stated until now. Since it is not so large, even if it performs count by processing by the processor, the whole engine performance is extent worsening 5%.

[0176] Moreover, although it is possible to process the CRC computation by software at a high speed several times compared with simple bit shift processing combining the table which stored the result of CRC beforehand calculated per octet, it is also possible to use this approach for count of a trailer part.

[0177] (2) The 1st term of a formula [0178]

[Equation 5]

$$(A' (X) X^{(L_B+L_T)}) \bmod G (X) \quad \dots (4)$$

Since $A' (X)$ is rewritten from $A (X)$ in *****, the count of a shift / XOR operation cannot be reduced using the result of (1). Therefore, the operation of the number-of-bits $(L_A+L_B+L_T)$ order of data is needed. But, the variable region of data is only 14 bytes of the head of $A' (X)$ in (4), and the remainder is the fixed value of the value 0 of bit (L_B+L_T) .

[0179] It is the data and L_B of $A' (X)$ at the time of initiation of an operation. The circuit which calculates the result of (4) by giving a number is L_B to a CRC count circuit. It is realizable by adding the circuit where only a time inputs the bit of a value 0. Naturally in this count, it is L_B+L_T . The operation of a time is required, and if it becomes $L_B+L_T \gg 14 \times 8$ (bit length of header unit $A' (X)$), the time amount of the almost same order as count of $V (X)$ will be taken, but as long as it sees from a processor, the value of (4) types is calculated in one data writing. Therefore, although the duration of count itself is order equivalent to the memory writing shown with the basic configuration of this example, and the method which performs CRC count to coincidence, since there is no need for bus access in the meantime, processing can be advanced in parallel to a processor, and the load of the processing to a processor and a bus becomes small.

[0180] If it is this way, it will become unnecessary moreover, not to evacuate the condition of a CRC count function, / to recover it, although the value of a CRC count function must be evacuated / recovered whenever it receives one cel when many connections are multiplexed.

[0181] (2) The 3rd term of a formula [0182]

[Equation 6]

$$T (X) \bmod G (X) \quad \dots (5)$$

It calculates by the ** CRC count function or software processing. The repeat order of this count is the 32 - 408 numbers of bits, and there is not. [so much]

[0183] The value of (2) types is calculated by taking the sum of the value of (3), (4), and (5) now.

[0184] Actual procedure is explained according to drawing 9 .

[0185] The following formula will be materialized if the Ethernet packet whole [1101], header units 1112-1114, and an information bureau 1115 are set to $C (X)$, $A (X)$, and $B (X)$, respectively.

[0186]

[Equation 7]

$$C(X) = A(X) X^{L_B} + B(X) \quad \dots (6)$$

Moreover, the following formula will be materialized if the AAL5 packet whole [1102], a header unit 1121, an information bureau 1123, and the trailer sections 1123-1127 are made into $C'(X)$, $A'(X)$, $B'(X)$, and $T'(X)$, respectively.

[0187]

[Equation 8]

$$C'(X) = A'(X) X^{(L_B + L_T)} + B(X) X^{L_T} + T(X) \quad \dots (7)$$

First, those CRC1301 and CRC1302 are calculated by a transfer and coincidence of parts for a header unit 1112-1114, and the information part 1115 at the processing steps S1311 and S1312. CRC of the Ethernet packet 1101 whole is calculated as those sums (1303) at step S1313.

[0188] CRC count of an AAL5 formal packet is performed as follows.

[0189] First, the CRC is calculated to the rewriting sexagenary-cycle coincidence of a packet head at step S1314. As already stated, the computational complexity requires only a part for the number of bits of a trailer (L_T of X^{**}) by using the result of CRC1302. The whole packet 1102 of AAL5 format is calculable by calculating CRC of the trailer parts 1123-1126 to addition and coincidence of a trailer at step S1316, and taking those sums at step S1317.

[0190] The approach of omitting the same CRC count of the same data explained with AAL5 of the <example of protocol conversion equipment between Frame Relay and ATM> above and an example of the protocol conversion equipment between Ether(s) can be extended also when the generating polynomials of CRC differ.

[0191] Below, the case where the count abbreviation approach of this CRC is applied to the Frame Relay of the basic configuration of this example mentioned above and IWF of ATM about the count abbreviation approach of CRC in the protocol conversion between the Frame Relais and ATM from which the method of CRC differs is explained.

[0192] (Invariable region) In drawing 5, the corresponding point 923 in the information field 913 and CPCS-PDU of a Frame Relay packet does not change before and after packet format conversion. The generating polynomial differs from the 16th order [32nd], respectively, and CRC calculated by each packet has it. [relatively prime]

[0193] First, CRC of these constant domains is calculable independently with other parts using the approach explained in the example 2.

[0194]

[Equation 9]

$$A(X) \bmod G1(X) \quad \dots (8)$$

$$A(X) \bmod G2(X) \quad \dots (9)$$

$A(X)$ ビット列913(923)の多項式表示

$G1(X)$ CRC16の生成多項式

$G2(X)$ CRC32の生成多項式

Now, two generating polynomials $G1(X)$ and product [of $G2(X)$] $G(X)$ are considered here. The degree of $G(X)$ is $16+32=48$. It asks for the remainder which broke $A(X)$ by $G(X)$.

[0195]

[Equation 10]

$$A(X) \bmod G(X) \quad \dots (10)$$

Since $G1(X)$ and $G2(X)$ are relatively prime, the following formula is materialized about (8) and (9).

[0196]

[Equation 11]

$$(A(X) \bmod G(X)) \bmod G_2(X) = A(X) \bmod G_1(X) = (7)$$

... (11)

$$(A(X) \bmod G(X)) \bmod G_1(X) = A(X) \bmod G_2(X) = (8)$$

... (12)

Thus, (9) and (10) can be calculated also by asking for a remainder by making the product of two generating polynomials into a generating polynomial, and breaking it by the original generating polynomial. Since the operation which breaks a remainder by the original generating polynomial is at most the 48th order in the degree order, i.e., this case, of $G(X)$, a processing load is very small.

[0197] Even if the degree of a generating polynomial becomes the 48th order from the 32nd order, the hardware scale of a CRC count function increases at most 1.5 times. Moreover, in calculating by the processor, if 64bitCPU is used, the processing number of steps will not change at all.

[0198] Thus, count of CRC can be communalized also about the case where generating polynomials differ.

[0199] Actual procedure is explained according to drawing 10.

[0200] The following formula will be materialized if the Frame Relay packet whole [912], a header unit 912, and an information bureau 913 are set to $C(X)$, $A(X)$, and $B(X)$, respectively.

[0201]

[Equation 12]

$$C(X) = A(X) X^{L_B} + B(X) \quad \dots (13)$$

Moreover, the following formula will be materialized if the AAL5 packet whole [912], a header unit 922, an information bureau 923, and the trailer sections 924-928 are made into $C'(X)$, $A'(X)$, $B'(X)$, and $T'(X)$, respectively.

[0202]

[Equation 13]

$$C'(X) = A'(X) X^{(L_B + L_T)} + B'(X) X^{L_T} + T'(X) \quad \dots (14)$$

$G(X) = G_1(X) \cdot G_2(X)$ and $G_1(X)$ take as the generating polynomial of CRC of a Frame Relay, and $G_2(X)$ is taken as the generating polynomial of CRC of AAL5.

[0203] First, the remainder 1401 by those $G(X)$ and a remainder 1402 are calculated by a transfer and coincidence of a part for a header unit 912, and the information part 91 at the processing steps S1411 and S1412. The remainder of the Frame Relay packet 921 whole is called for as those sums (1403) at step S1413.

[0204] CRC count of an AAL5 formal packet is performed as follows.

[0205] First, the CRC is calculated to the rewriting sexagenary-cycle coincidence of a packet head at step S1415. And in CRC count of the information part 923 in step S1316, the result (remainder 1402) obtained at said processing step S1412 is used like the 2nd example. The remainder 1408 by $G(X)$ of the packet 1102 whole of AAL5 format is calculable by calculating CRC of the trailer parts 924-928 to addition and coincidence of a trailer at step S1417, and taking those sums at step S1318.

[0206] CRC of a Frame Relay is calculated by asking for the remainder which broke 1408 by the remainder $G_1(X)$ by $G(X)$ at step 1419.

[0207] This invention is not limited to each example mentioned above, it is the range which does not deviate from the summary, and can deform variously and can be carried out.

[0208]

[Effect of the Invention] In this invention, when a duplication part exists in the transmission information before and behind protocol conversion, since error detecting code computation performed before and after protocol conversion is performed only at once before and after conversion to this duplication part, it can expect improvement in the speed of error detecting

code computation, therefore improvement in the speed of protocol conversion processing.
 [0209] Moreover, when performing error detecting code computation by processor processing, reduction of processing loads can be aimed at by [which is a processing load] coming size and substituting error detecting code computation only for once to this duplication part.

[Translation done.]

*** NOTICES ***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] Drawing showing the fundamental hardware configuration concerning one example of this invention

[Drawing 2] Drawing for explaining the concept of the process in this example

[Drawing 3] Drawing showing the relation between the process and circumference hardware in ATM-FR of this example, and data flow

[Drawing 4] Drawing showing the relation between the process and circumference hardware in FR-ATM of this example, and data flow

[Drawing 5] Drawing showing the packet format of ATM (AAL5) in this example, and a Frame Relay

[Drawing 6] Drawing showing an example of the hardware configuration which used the 2 port RAM

[Drawing 7] Drawing showing the packet format of Ethernet and ATM

[Drawing 8] Drawing showing an example of the hardware configuration of the protocol conversion equipment between Ethernet-ATM

[Drawing 9] Drawing which shows the procedure of CRC count -- a chart

[Drawing 10] Drawing which shows other procedure of CRC count -- a chart

[Drawing 11] Interface conceptual diagram

[Drawing 12] Protocol conceptual diagram

[Drawing 13] Drawing for explaining software / functional partition therefore realized hard

[Drawing 14] Drawing showing the flow of the processing in protocol conversion equipment

[Description of Notations]

501 -- A processor, 502 -- A processor bus, 503 -- Processor memory, 504 -- An output cel buffer, 505 -- An input cel buffer, 506 -- Frame Relay physics layer processing facility, 507 -- A Frame Relay physics layer processing facility, 508 -- CRC count function, 509 -- An ATM switch, 601 -- Receiving-side ATM layer ability process, 602 -- An ATM-Q.922 conversion functional process, 6031 -- Q.922-ATM conversion functional process, 604 -- A transmitting-side ATM layer ability process, 605 -- OAM functional process, 712 -- A cel header-function conversion table, 721 -- Empty cel processing facility subroutine, 722 -- An OAM cel processing facility subroutine, 723 -- AAL (SAR) functional subroutine, 732 -- An OAM cel buffer, 741 -- ATM-FR conversion functional process, 742 [-- Buffer,] -- An AAL buffer, 751 -- A buffer, 752 -- A buffer, 753 812 -- A connection-function conversion table, 821 -- OAM cel processing facility subroutine, 822 -- An output cel display, 823 -- AAL (SAR) functional subroutine, 824 --

An output cel display, 832 -- An OAM cel buffer, 841 -- FR/ATM conversion functional process, 842 [-- Buffer,] -- An AAL buffer, 851 -- A buffer, 852 -- A buffer, 853 1001 -- A processor, 1002 -- A processor bus, 1003 -- Receiving memory, 1004 -- Transmitting memory, 1005 -- An output cel buffer, 1006 -- Input cel buffer, 1007 -- Receiving-side FR-PHY, 1008 -- Transmitting-side FR-PHY, 1010 -- ATM switch, 1201 [-- An output cel buffer 1205 / -- An input cel buffer, 1206 / -- An Ethernet interface, 1208 / -- ATM switch] -- A processor, 1202 -- A processor bus, 1203 -- Buffer memory, 1204

[Translation done.]

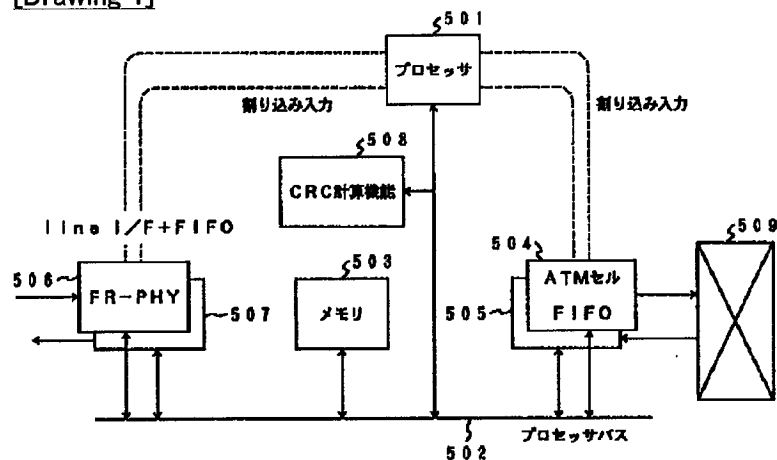
* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

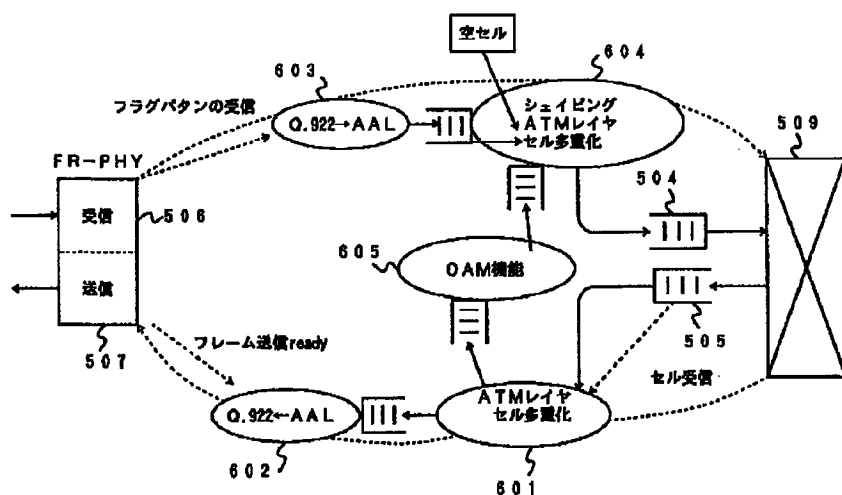
- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DRAWINGS

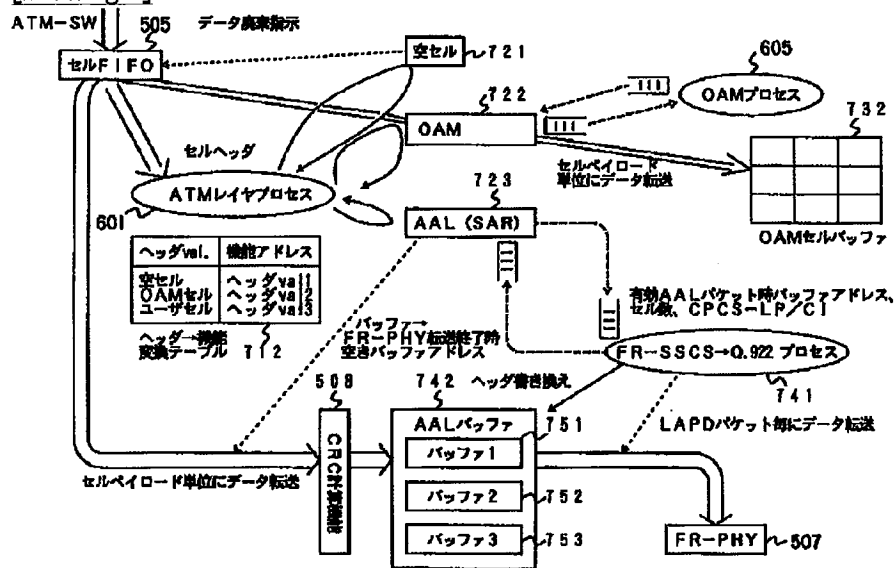
[Drawing 1]



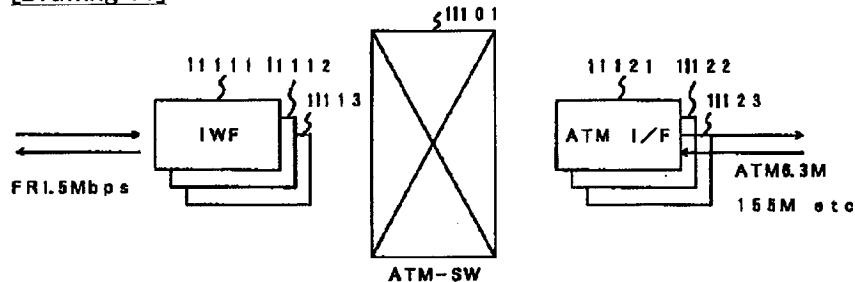
[Drawing 2]



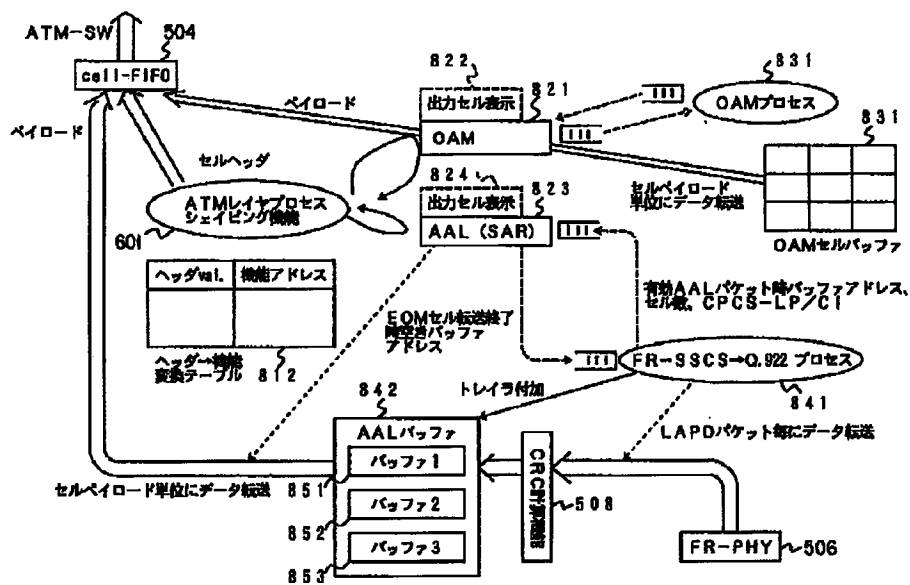
[Drawing 3]



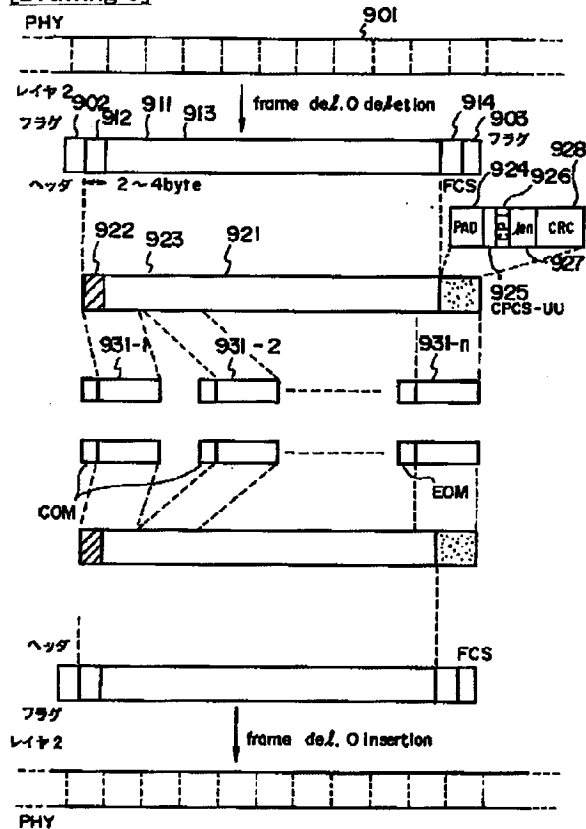
[Drawing 11]



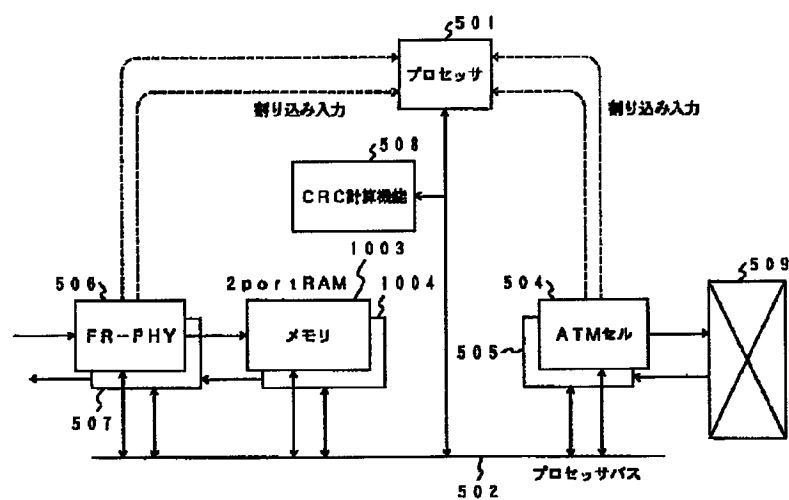
[Drawing 4]



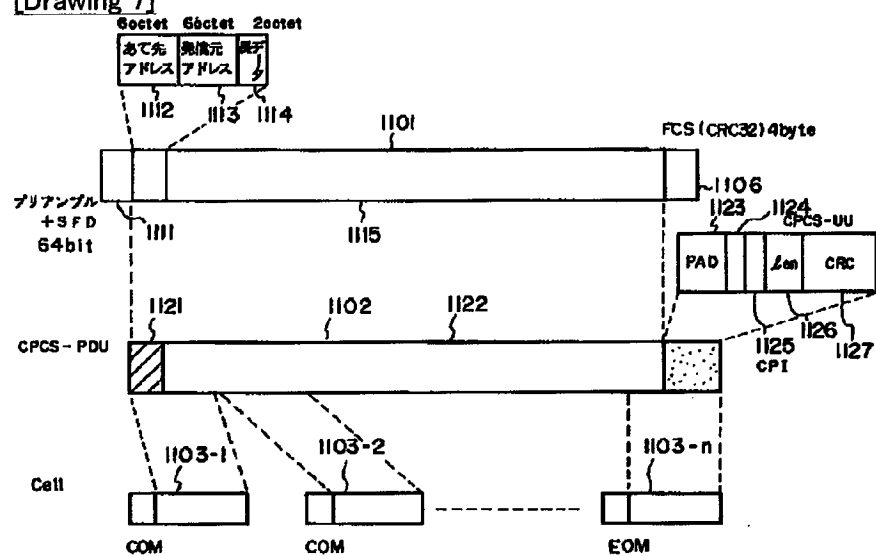
[Drawing 5]



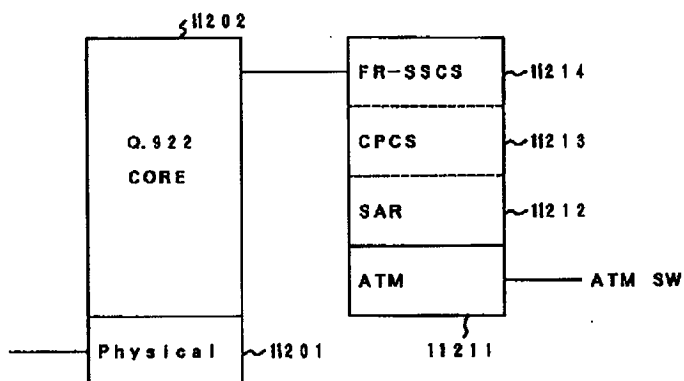
[Drawing 6]



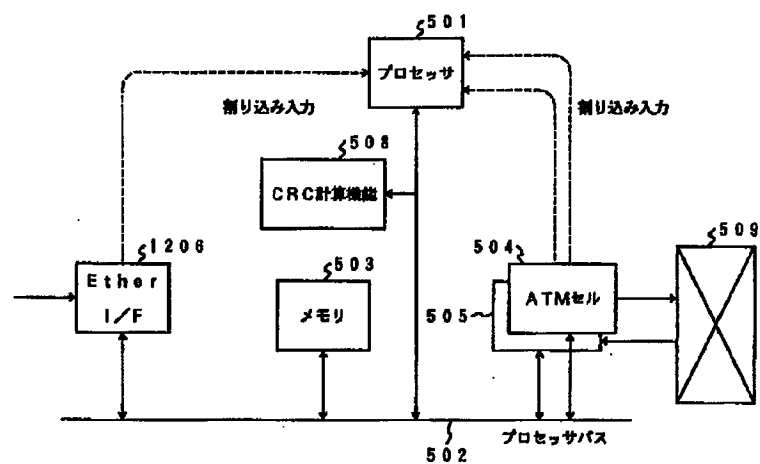
[Drawing 7]



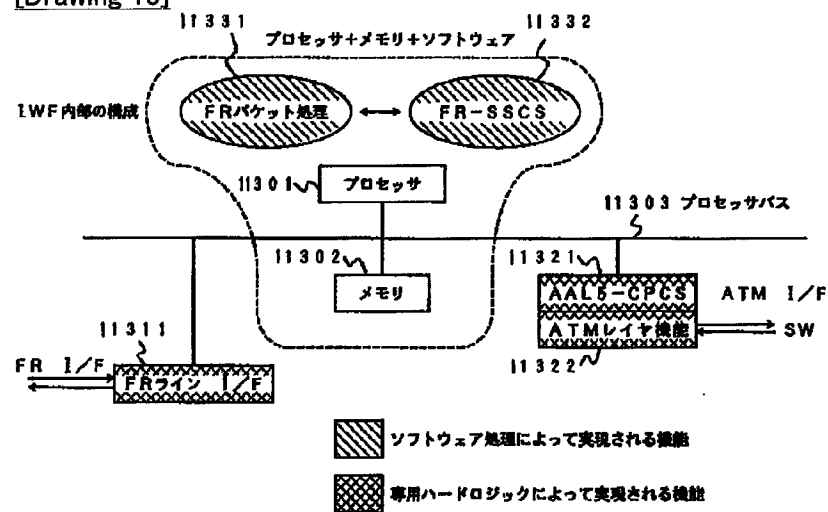
[Drawing 12]



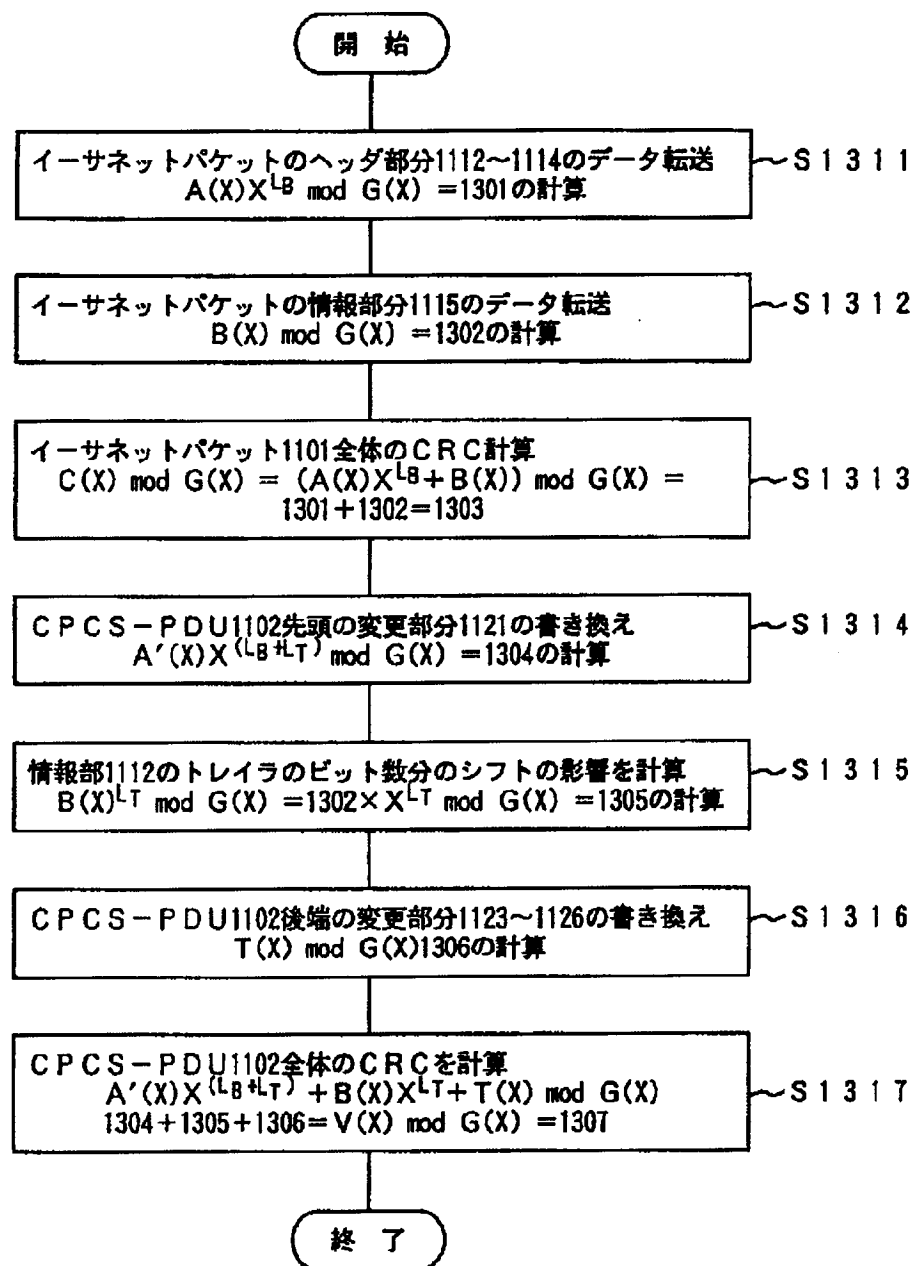
[Drawing 8]



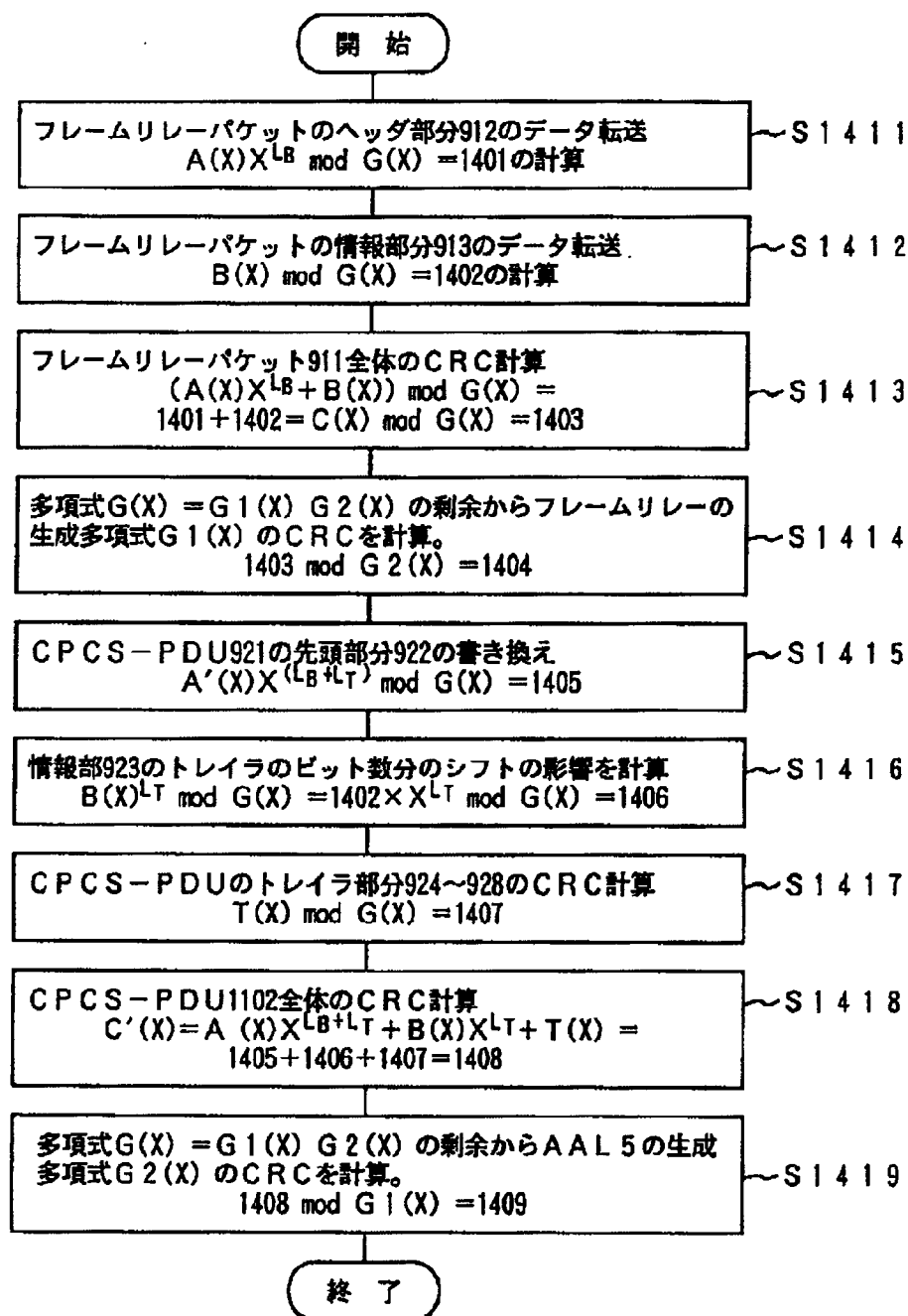
[Drawing 13]



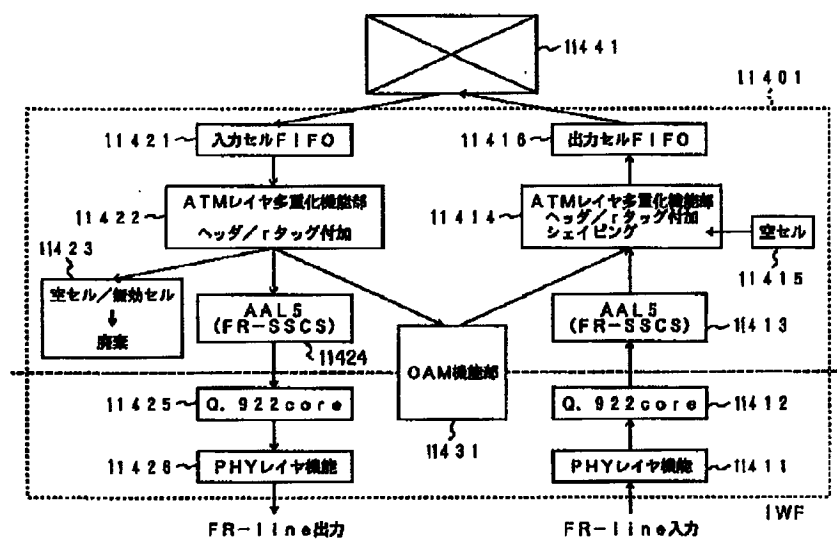
[Drawing 9]



[Drawing 10]



[Drawing 14]



[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-102761

(43) 公開日 平成8年(1996)4月16日

(51) Int.Cl.⁶

識別記号

片内整理番号

F I

技術表示箇所

H 0 4 L 12/64
12/28
29/06

9466-5K

H 0 4 L 11/ 20

A

9466-5K

E

審査請求 未請求 請求項の数 4 F D (全 25 頁) 最終頁に続く

(21) 出願番号 特願平6-261280

(22) 出願日 平成6年(1994)9月30日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 橋本 幹生

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(72) 発明者 角田 啓治

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(72) 発明者 正畑 康郎

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

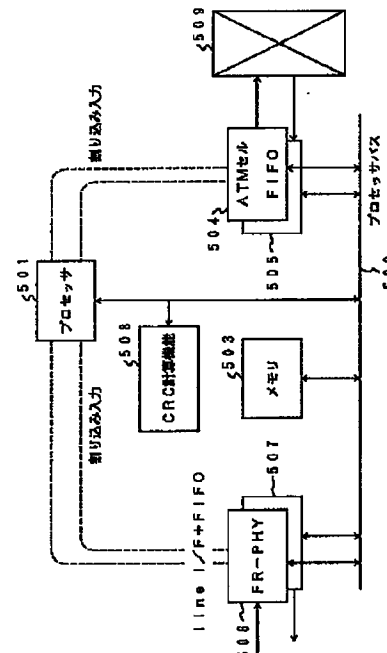
(74) 代理人 弁理士 鈴江 武彦

(54) 【発明の名称】 プロトコル変換装置およびプロトコル変換方法

(57) 【要約】

【目的】 誤り検出符号計算処理を高速化したプロトコル変換装置を提供すること。

【構成】 ATMプロトコルに従う第1の伝送情報とATMプロトコル以外のプロトコルに従う第2の伝送情報の相互間で、該伝送情報に対するプロトコル変換処理を行うとともに、該プロトコル変換処理のうち誤り検出処理を含む少なくとも一部の処理を少なくとも1つのプロセッサを用いて処理するプロトコル変換装置において、前記第1及び第2の伝送情報に対して、プロトコル変換前のプロトコルにおける誤り検出処理を施す手段と、前記第1及び第2の伝送情報のうちプロトコル変換にて内容が変化した部分に対して、変換後のプロトコルにおける誤り検出処理を施す手段とを具備したことを特徴とする。



【特許請求の範囲】

【請求項1】第1のプロトコルに従う第1の伝送情報を入力し、該第1の伝送情報を第2のプロトコルに従う第2の伝送情報にプロトコル変換するプロトコル変換装置において、

入力した前記第1のプロトコルに従う第1の伝送情報に対して、該第1のプロトコルにおける誤り検出処理を、少なくとも1つのプロセッサ内で処理する第1の誤り検出処理手段と、

前記第2のプロトコルへとプロトコル変換された前記第2の伝送情報のうち、該プロトコル変換後に内容が変化する部分に対して、該第2のプロトコルにおける誤り検出処理を、少なくとも1つのプロセッサ内で処理するとともに、該プロトコル変換後に内容が変化しない部分に対しては、前記第1の誤り検出処理手段による処理結果を利用する第2の誤り検出処理手段とを具備したことを特徴とするプロトコル変換装置。

【請求項2】ATMプロトコルに従う第1の伝送情報とATMプロトコル以外のプロトコルに従う第2の伝送情報の相互間で、該伝送情報に対するプロトコル変換処理を行うとともに、該プロトコル変換処理のうち誤り検出処理を含む少なくとも一部の処理を少なくとも1つのプロセッサを用いて処理するプロトコル変換装置において、

前記第1及び第2の伝送情報に対して、プロトコル変換前のプロトコルにおける誤り検出処理を施す手段と、

前記第1及び第2の伝送情報のうちプロトコル変換にて内容が変化した部分に対して、変換後のプロトコルにおける誤り検出処理を施す手段とを具備したことを特徴とするプロトコル変換装置。

【請求項3】ATMプロトコルに従う第1の伝送情報とATMプロトコル以外のプロトコルに従う第2の伝送情報の相互間で、該第1及び第2の伝送情報に対して相互間でプロトコル変換処理を行うとともに、該プロトコル変換処理のうち誤り検出処理を含む少なくとも一部の処理を少なくとも1つのプロセッサを用いて行うプロトコル変換装置において、

プロトコル変換前後の前記伝送情報中に重複部分が存在する場合、該重複部分の誤り検出処理を変換前後において一度だけ行う手段を具備したことを特徴とするプロトコル変換装置。

【請求項4】ATMプロトコルに従う第1の伝送情報とATMプロトコル以外のプロトコルに従う第2の伝送情報の相互間で、該伝送情報に対するプロトコル変換処理を行うプロトコル変換処理方法において、

いずれかの前記プロトコルに従う前記伝送情報を入力するステップと、

前記伝送情報に基づき所定の誤り検出処理を行うとともに、該伝送情報のフォーマット変換を行うステップと、

前記伝送情報のうちフォーマット変換によって内容が変

化された部分に対して、変換後のプロトコルにおける誤り検出処理を施すステップとを有することを特徴とするプロトコル変換方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ATMとATM以外のインタフェース形式の通信を相互接続する装置および方法に係り、それぞれのインタフェース固有の形式で送信されるデータを相互に変換するプロトコル変換装置および方法に関する。

【0002】

【従来の技術】2つの通信方式の間で情報を互いに転送するために設けられる装置は、プロトコル変換装置（Inter Working Functionと呼ばれる。ここではATMとフレームリレーとの間を結ぶプロトコル変換装置（以下、IWFと呼ぶ）を例としてATMインタフェースを持つ従来のIWFの技術を説明する。

【0003】フレームリレーは、ITU-T勧告X.25に代表される既存パケット網を高速化する場合に有力なプロトコルである。フレームリレー（以下、FRと略す）では、通信路の伝送品質向上にともなって、リンク毎に行っていた再送制御などのプロトコルを簡略化した結果、既存パケット通信に近い利用者インタフェースを持ちながら、通信速度、遅延性能が向上している。

【0004】一方、ATMは音声、データなどの各種通信を統合するプロトコルである。ATMでは、情報はセルという比較的短いパケットに格納され、ルーティングされる。

【0005】図11に、交換装置の中に実装された場合のIWFの概念図を示す。このIWFは、ATMとFRの2種類のインタフェースを持つ。

【0006】ATMインタフェース11121～11123から入力されたセルは、ATMスイッチ11101でスイッチングされ、そのうちの予め定められたコネクションのセルはIWF11111～11113に入力される。IWFはセルの形で入力されたパケットをFR形式に変換し、その他必要な処理を行ってFRのラインインタフェースに出力する。IWFは、FRのインタフェースから入力されたパケットをセルに分解し、ATMへと送出する処理も行う。

【0007】IWF機能仕様については、ITU-T勧告I.365.1 Specific Convergence Sublayer (FR-SSCS) between Frame Relaying and B-ISDN)に定められている。仕様上、IWFは単純にそれぞれのプロトコル機能(Q.922 core, FR-SSCS (Frame Relaying Service Specific Convergence Sublayer))を接続することによって、例えば

図12に示すように簡単に実現可能である。

【0008】ところで、従来、2つのプロトコル機能は、それぞれ次のような方法で実装されてきた。

【0009】(ハード/ソフトの分担構成例) 図13は、従来のIWF内部の機能配置の概念図である。図13において、11301、11302、11303は、それぞれプロセッサ、メモリ、プロセッサバスを表す。11311、11321、11322は、それぞれハードウェアによって構成されるFR-PHY部、AAL5-CPCS部、ATMレイヤ機能である。また、11332、11332は、それぞれプロセッサによって実行されるフレームリレープロトコル処理(以下Q.922処理と略す)、FR-SSCS処理である。

【0010】フレームリレープロトコルの処理機能のうち、レイヤ1とレイヤ2の一部に属する機能は、ITU-T勧告I.431に従うレイヤ1物理レイヤ処理を論理ゲートによって構成した専用ハードウェア11311として実現される。また、ITU-T勧告Q.922コアに従うレイヤ2処理は、プロセッサによるソフトウェア処理11331として実装される。

【0011】また、ATM側の機能のうち、ATMレイヤ機能とAAL5-CPCS部の機能は、専用ハードウェアとして実装される。そして、AALのサービス固有部、FR-SSCSは、プロセッサによるソフトウェア処理として実装される。

【0012】ATMレイヤ機能からCPCS機能までが、専用ハードウェアによって実装されるのは、高速なATMインタフェースに対応する場合にはソフトウェア処理では必要な速度が出ないためである。

【0013】(処理フローの概要) 次に、図14に基づいてIWFで行われる処理の内容をより詳細に説明する。図14において、11411、11412、11413、11414、11415、11416は、フレームリレーインタフェースからの入力を処理する機能群であり、それぞれ物理レイヤ機能部、Q.922機能部、AAL5機能部、ATMレイヤ機能部、シェイピング(空セル挿入)機能部、出力セルバッファを表す。また、11421、11422、11423、11424、11425、11426は、ATMスイッチ11441からの入力を処理する機能であり、それぞれ入力セルバッファ、ATMレイヤ多重化機能部、空セル/無効セル処理機能部、AAL5機能部、Q.922機能部、物理レイヤ機能部を表す。11431は、入出力共通のOAM機能部を表す。

【0014】(ATMからFRへの変換) ATMスイッチ11441から入力したATMセルは、ITU-T勧告I.361に記述されている仕様に基づいて、ATMレイヤ機能部11422でコネクション毎に分離される。セルの分離に先だって、ATMスイッチ11441内のルーティングやOAMのために付加されたルーティ

ングタグ情報は削除される。

【0015】OAMセルは、OAM機能部11431へ送られ、空セル/無効セルはそれぞれのセルを扱う機能によって廃棄される。予め定められたAALコネクションに対応するヘッダ値を持つセルは、AAL機能部11424に渡され、当該AAL機能部11424によって処理される。セルを受信したAAL機能部11424のSARサブレイヤでは、ITU-T勧告I.363に記述されている仕様に基づいて、セルヘッダ情報を参照しながら、複数のセルペイロードをつなぎ合わせて、SAR-SDUを再構成する。そして、AALのCPCSサブレイヤ機能部で誤り検出などの処理が行われる。これらの処理をハードウェア的に行う方法は、広く知られ実用化されている。図13の構成図に示したATMレイヤ機能部11322、AAL5-CPCS機能部11321がこれに対応する。

【0016】図13の構成例では、次の処理はプロセッサ11301によって行われる。CPCSサブレイヤのバッファメモリに格納されたCPCS-SDUは、プロセッサによって読み出されてプロセッサのバッファメモリ(図示せず)に転送される。AAL機能のうちFR-SSCSは、CPCSサブレイヤから受けとったCPCS-SDUのヘッダのフィールドを一部書き換えることだけなので、バッファメモリ上のデータを操作することにより容易に実現可能である。そして、今度はこのパケットデータに、Q.922機能部11425の処理が行われる。ここでの処理は、論理チャネルによる多重化、DLCIの付け換えである。FR-SSCSから渡されたパケットデータのDLCIフィールドを、予めその値毎に定められた書き換えDLCI値に書き換える。

【0017】以下の処理は、図13の構成例ではFR-PHY11311によって行われる。まず、レイヤ2パケットについて、CRC-16の計算手順に基づいてCRCが計算され、パケットの末端に付加される。そして、フレームの前後に、フラグボタンが付加される。このデータ内容にフラグボタンと同じボタンが出現することを防ぐために、1bit単位で値0のビットデータが挿入され、このデータに本来の物理レイヤ(レイヤ1)処理すなわちラインコーディングの処理が行われて、フレームリレーインタフェースのライン出力としてデータが送出される。

【0018】(FRからATMへの変換) FRからATMへの変換も、ほぼ同じ手順をたどる。多少異なるのは、ATMレイヤにおいて送信コネクションが予め定められた送信レートを越えないように、適宜他のコネクションやOAMのセルを多重化しながら、あるいはそれで不十分な場合は強制的に空セルを挿入することによって、送信コネクションの送信レートを規定値以下に押えるシェイピング機能部11415があることである。

【0019】このようにして従来からあるATMやフレ

10

20

30

40

50

ームリレーの専用ハードウェアを組み合わせることによってIWFを実現することが可能である。

【0020】ところで、これらの処理を実装する上では、必ずしも専用ハードウェアが必須とは限らない。汎用プロセッサによる通信プロトコル処理の方法を用いればATMレイヤ、AAL、FR物理レイヤのプロセッサによる処理が実用に耐える速度が達成できる可能性がある。従来専用ハードウェアによって実現されていた機能を高性能汎用プロセッサのソフトウェア機能によって実現することにより、専用ハードウェアを開発するコストを削減するのみならず、汎用プロセッサ部分の量産化によるコストダウンも期待できる。

【0021】既に説明したように、ATM-フレームリレーIWFをソフトウェア処理によって実現すること自体は、従来技術を単純に組み合わせることによって容易に実現可能である。だが、この実装方法は、プロセッサ処理の特性に合わせた最適化の余地を残しており、プロセッサ処理に適した処理方式をとることによって性能の向上もしくはコストの引下げを図ることが可能である。また、ATMレイヤ機能/AAL機能をソフトウェア処理によって実現することにより、将来別のATMレイヤ機能やAAL機能に基づくサービスが現れても、既存のIWFのハードウェアはそのままに対応が可能になる。

【0022】ところが、ATM-FRインタワーキングにおけるソフトウェアによる通信機能の高速化は困難であった。これは、1つにはプロセッサ処理固有の問題点に起因している。

【0023】プロセッサ処理固有の問題として、まず、プロセッサ-メモリ間転送のオーバーヘッドが挙げられる。近年の高性能プロセッサの速度向上はめざましいが、一方で周辺記憶装置の速度向上がプロセッサに追いつかないため、メモリアクセスが処理速度の制限要因になる、いわゆるvon neumann bottleneckの現象が顕著になって生じている。とりわけ、次々と到着する新しいデータを処理しなければならない通信装置で、この問題が顕著に現れることが指摘されている(D. Banks, Mprudence, "A

High Performance Network Architecture for aPA-RISC Workstation". IEEE Journal on Selected Areas in Communication 1993 vol. 10 No. 1)。複雑なデータの操作ではなく、主記憶のある部分から別の部分への単なるコピーのために、プロセッサが長時間使用されてしまうのである。特に、階層型プロトコルをそのまま実装した場合、プロトコルの階層毎にデータのコピーを行うことは、大変なプロセッサ能力とプロセッサバス転送能力の浪費である。

【0024】従来からある解決法として、DMA転送によってプロセッサの動作と並行してデータ転送を行うこ

とも可能である。だが、プロセッサと記憶装置を共有して並行処理を行うためには、複雑な競合制御の機構が必要である。また、複雑な機構には、それ自身にオーバーヘッドが発生し、コピーを並行処理しただけの性能の向上が見込めない場合も多い。さらに、複雑な機構は、コスト増を招いてしまう。近年の高性能汎用プロセッサでは、パイプラインと呼ばれる命令操作の並列実行によって、メモリ-メモリ間転送においてDMA転送と同等の性能を発揮するものがある。もちろん、プロセッサを使ってデータ転送を行っている間はプロセッサによるデータ処理はできないが、これはDMA装置をとり入れることとの間のトレードオフ関係であり、現在の技術においてはデータ転送の速度はプロセッサを使用する場合もDMA装置を使用する場合も大差ない。プロセッサ自身のデータ転送能力を有効に利用すれば、バスマスタがプロセッサのみの単純なハードウェア構成で比較的高速なデータ転送処理を行うことができる。さらに、従来のプロトコル処理にあった不要なデータ転送を減らすことが必要となる。

【0025】データ転送の問題とは別に、プロセッサ処理には、不向きな処理がある。例えば、単純なビット操作を数多く繰り返す、スクランブル/デスクランブル処理や、CRCの計算処理である。プロセッサは、ワード幅のデータを効率よく処理することはできるが、ワードに跨ったデータの処理やワード内のビットフィールドに対する繰り返し処理の効率は良くないためである。

【0026】

【発明が解決しようとする課題】従来のプロトコル変換装置および方法では、誤り検出符号の計算処理を効率良くかつ高速に行うことはできなかった。また、誤り検出符号の計算をプロトコル変換前と変換後の2つのパケット形式のそれぞれについて計算しなければならなかったため、変換後も変わらない部分の計算については無駄が生じており、処理速度の向上を妨げるものとなっていた。

【0027】本発明は、上記事情に鑑みてなされたものであり、誤り検出符号計算処理を高速化したプロトコル変換装置および方法を提供することを目的とする。

【0028】

【課題を解決するための手段】本発明は、第1のプロトコルに従う第1の伝送情報を入力し、該第1の伝送情報を第2のプロトコルに従う第2の伝送情報にプロトコル変換するプロトコル変換装置において、入力した前記第1のプロトコルに従う第1の伝送情報に対して、該第1のプロトコルにおける誤り検出処理を、少なくとも1つのプロセッサ内で処理する第1の誤り検出処理手段と、前記第2のプロトコルへとプロトコル変換された前記第2の伝送情報のうち、該プロトコル変換後に内容が変化する部分に対して、該第2のプロトコルにおける誤り検出処理を、少なくとも1つのプロセッサ内で処理すると

ともに、該プロトコル変換後に内容が変化しない部分に対しては、前記第1の誤り検出処理手段による処理結果を利用する第2の誤り検出処理手段とを具備したことを特徴とする。

【0029】また、本発明は、ATMプロトコルに従う第1の伝送情報とATMプロトコル以外のプロトコルに従う第2の伝送情報の相互間で、該伝送情報に対するプロトコル変換処理を行うとともに、該プロトコル変換処理のうち誤り検出処理を含む少なくとも一部の処理を少なくとも1つのプロセッサを用いて処理するプロトコル変換装置において、前記第1及び第2の伝送情報に対して、プロトコル変換前のプロトコルにおける誤り検出処理を施す手段と、前記第1及び第2の伝送情報のうちプロトコル変換にて内容が変化した部分に対して、変換後のプロトコルにおける誤り検出処理を施す手段とを具備したことを特徴とする。

【0030】また、本発明は、ATMプロトコルに従う第1の伝送情報とATMプロトコル以外のプロトコルに従う第2の伝送情報の相互間で、該第1及び第2の伝送情報に対して相互間でプロトコル変換処理を行うとともに、該プロトコル変換処理のうち誤り検出処理を含む少なくとも一部の処理を少なくとも1つのプロセッサを用いて行うプロトコル変換装置において、プロトコル変換前後の前記伝送情報中に重複部分が存在する場合、該重複部分の誤り検出処理を変換前後において一度だけ行う手段を具備したことを特徴とする。

【0031】また、本発明は、ATMプロトコルに従う第1の伝送情報とATMプロトコル以外のプロトコルに従う第2の伝送情報の相互間で、該伝送情報に対するプロトコル変換処理を行うプロトコル変換処理方法において、いずれか前記プロトコルに従う前記伝送情報を入力するステップと、前記伝送情報に基づき所定の誤り検出処理を行うとともに、該伝送情報のフォーマット変換を行うステップと、前記伝送情報のうちフォーマット変換によって内容が変化した部分に対して、変換後のプロトコルにおける誤り検出処理を施すステップとを有することを特徴とする。

【0032】

【作用】本発明では、プロトコル変換前後の伝送情報中に重複部分が存在する場合、プロトコル変換の前後で行う誤り検出符号計算処理（例えばCRC計算処理）は、該重複部分に対しては変換前後において一度だけ行う。

【0033】例えばATMとフレームリレーの相互間でのプロトコル変換では、どちらの形式のケットにも誤り検出のためCRC符号が付加される。プロトコル変換の過程で書換えられたり付加／削除される領域はごく僅かなので、この2種類のケットは大半のデータ領域が重複部分となる。

【0034】従来、2種類のインタフェースを処理する機能を単純に結合していたので、この共通データ部分

なわち重複部分に誤り検出符号計算処理を2度行っていたのに対し、本発明では、データ領域の大半をしめる重複部分に対して、誤り検出符号計算処理は変換前後において一度だけ行うので、プロトコル変換処理の高速化が期待できる。

【0035】また、誤り検出符号計算処理をプロセッサ処理にて行う場合、処理負荷の大きい誤り検出符号計算処理を該重複部分に対しては一度だけで済ますことにより、処理負荷の削減を図ることができる。

【0036】

【実施例】以下、図面を参照しながら本発明の実施例を説明する。

【0037】＜実施例の基本構成＞はじめに、本発明の一実施例の基本構成・動作を説明する。

【0038】本実施例では、プロトコル変換装置（IWF装置）にプロセッサ処理を適用する場合の基本構成について説明する。以下、図1～6に基づいて本実施例を説明する。

【0039】図1において、501はプロセッサ、502はプロセッサバス、503はプロセッサメモリ、504はATMセルをATMスイッチに出力するための出力セルバッファ、505はATMスイッチからATMセルを入力するための入力セルバッファ、506はフレームリレーライン入力を受信してレイヤ2ケットを生成するフレームリレー物理レイヤ処理機能（以下FR-PHYとする）、507はレイヤ2ケットを入力してフレームリレーラインへ出力を行うフレームリレー物理レイヤ処理機能（以下FR-PHYとする）、508はプロセッサがメモリヘッダを書き込むと同時にそのCRCを計算するCRC計算機能、509はATMスイッチである。

【0040】図2において、601は受信側ATMレイヤ機能プロセス、602はATMセルをケットに組み上げてフレームリレー形式に変換する、AAL5（FR-SSCS）とQ.922プロトコルの一部を処理するATM-Q.922変換機能プロセス（以下ATM-FR変換機能プロセスと呼ぶ）、603はフレームリレーケットを変換してATMセルに分解するQ.922プロトコルの一部とAAL5（FR-SSCS）を処理するQ.922-ATM変換機能プロセス（以下FR-ATM変換機能プロセスと呼ぶ）、604は送信側ATMレイヤ機能プロセス、605はOAM機能プロセス、504は入力セルバッファ、505は出力セルバッファ、507は出力FR-PHY、506は入力FR-PHY、509はATMスイッチである。

【0041】図3は、ATM-FR変換におけるプロセッサの入出力装置、各機能プロセスとデータとの関係を示している。

【0042】図3において、505は入力セルバッファ、601はATMレイヤプロセス、712はセルヘッ

ダー機能変換テーブル、721はATMレイヤプロセスから呼ばれる空セル処理機能サブルーチン、722はATMレイヤプロセスから呼ばれるOAMセル処理機能サブルーチン、723はATMレイヤプロセスから呼ばれるAAL(SAR)機能サブルーチン、605はOAMプロセス、732はOAMセルバッファ、741はATM-FR変換機能プロセス、742はAALバッファ、508はCRC計算機能、751はAALバッファ741の中にあるバッファ1、752はAALバッファ741の中にあるバッファ1、753はAALバッファ741の中にあるバッファ1、507は出力FR-PHYである。

【0043】図4は、FR-ATM変換におけるプロセッサの入出力装置、各機能プロセスとデータとの関係を示している。

【0044】図4において、504は入出力セルバッファ、601は出力ATMレイヤプロセス、812はコネクション機能変換テーブル、821はATMレイヤプロセスから呼ばれるOAMセル処理機能サブルーチン、822はATMレイヤプロセスに対してOAM出力データが存在することを示す出力セル表示、823はATMレイヤプロセスから呼ばれるAAL(SAR)機能サブルーチン、824はATMレイヤプロセスに対してAAL出力データが存在することを示す出力セル表示、605はOAMプロセス、832はOAMセルバッファ、841はFR-ATM変換機能プロセス、842はAALバッファ、508はCRC計算機能、851はAALバッファ841の中にあるバッファ1、852はAALバッファ841の中にあるバッファ2、853はAALバッファ841の中にあるバッファ3、506は入力FR-PHYである。

【0045】図5は、IWFで処理される各パケットの形式を示している。

【0046】図5において、901はフレームリレー伝送線を伝送される形式の信号を表す。902、903はフラグボタン、911はレイヤ2フレーム、912~914はそれぞれレイヤ2フレームを構成するヘッダ、情報フィールド、FCS、921はCPCS-PDU、922~923はそれぞれCPCS-PDUを構成する情報フィールド、トレイラ、924~928はそれぞれトレイラを構成するPAD、CPCS-UU、CPI、Length、CRC、931-1、931-2~931-nはセルである。

【0047】図6に、2port RAMを用いた場合の本実施例の構成を示す。

【0048】1001はプロセッサ、1002はプロセッサバス、1003は受信メモリ、1004は送信メモリ、1005は出力セルバッファ、1006は入力セルバッファ、1007は受信側FR-PHY、1008は送信側FR-PHY、1009はCRC計算機能、10

10はATMスイッチである。

【0049】以下、本実施例をさらに詳細に説明する。

【0050】本実施例においては、メモリという語句は、通常のプロセッサメモリであるところのメインメモリではなく、現在の高速プロセッサにおける「キャッシュメモリ」を示すものとする。高速キャッシュメモリと比較的低速なメインメモリの併用によって、高速化とメモリの大容量化ができることはよく知られているが、通信プロトコル処理の場合、キャッシュ機構が効率よく動作する条件すなわちアクセスの局所性が少なくともデータについては成立しない。また、通信プロトコル処理の場合、内部にタイマを持つため、命令のスワップアウトを行うことによる実行時間の不確定性は、実装を困難にする要因である。よって、ここでは命令/データともキャッシュに括りつけて実行することを考える。また、プロセッサ周辺装置も、いわゆるプロセッサバスではなくキャッシュメモリのバスもしくは同等の高速バスに接続することを前提としている。これは、メモリアクセス時にプロセッサ中のパイプライン動作を乱さないよう、FI

【0051】次に、本実施例の前提となるプロセッサ処理によるATM-FR-IWF装置の基本動作を、ATM-FR変換、FR-ATM変換の順序で説明する。

【0052】〔概略動作〕図1及び図2に基づいて、本発明の基本となるハードウェアとプロセッサ上のプロセスの動作の概要を説明する。

【0053】まず、ATMスイッチ509から入力したセルは、入力セルバッファ505に蓄積される。この内容は、プロセッサ501によって読み出され、以降はプロセッサ501上のプロセスで扱われる。

【0054】ATMレイヤプロセス601は、セルヘッダの値に基づいて予め定められた処理を起動する。フレームリレー対応のコネクションのセルは、プロセッサメモリ503に転送され、セルに分割されたパケットが再構成される。再構成されたパケットデータは、ATM-FR変換プロセス602に渡されて、ATMとフレームリレーの変換に必要な処理が施される。ここまででプロセッサ501上の処理が終了、パケットはFR-PHY507に引き渡されて、ラインインタフェースに送信される。

【0055】受信側では、FR-PHY506でレイヤ2パケットが検出されると、プロセッサ501上のFR-ATM変換プロセス603が起動され、データはプロセッサメモリ503に転送されて、パケットがAAL5(FR-SSCS)形式に変換される。このパケットは、ATMレイヤプロセス604でセルに分割され、ATMレイヤプロセス604の一機能であるシェイピング機能によってコネクション毎に予め定められたセルレートを越えないようにして、セル流として出力セルバッファ504に書き込まれ、ATMスイッチ509へ出力さ

れる。

【0056】次に、図3及び図4に基づいてプロセッサメモリ上のデータ構造を含めて詳細に動作を説明する。

【0057】[ATMからFRへの変換]

(セルの入力 ATMプロセスの起動タイミング) ATMスイッチから出力されたセルは、FIFOバッファ505に格納される。FIFOバッファ505に蓄積されたセルの量が予め定められた一定値を越えると、プロセッサに割り込み信号が入力される。この割り込み信号によって、受信側ATMレイヤプロセス711が起動される。セルの到着をプロセッサへ通知することは、割り込み信号によらず、例えばプロセッサからの定期的なポーリングによっても可能ではあるが、一般に割り込み処理はポーリングよりも応答が高速かつ無駄な監視処理が不要なので、プロセッサの処理負荷が少いため、割り込み処理の方が有利である。

【0058】(セル入力 バイト単位のアラインメント) バッファにセルを格納する際には、プロセッサの扱いを容易にするため、ワード単位でアラインメントを取ることが望ましい。例えば1ワードが8バイトで構成されるバスインタフェースにおいては、セルの先頭に3バイトの付加部分(PAD)を設けて、1セルのヘッダがワードの中で常に同じように位置するように格納する方法が知られている。一般に、スイッチでセルを扱う場合には、53byteのセルに数byteのルーティングタグを付加して扱うことが多い。11byteのルーティングタグを付加することによって、セル長を64byteとしてアラインメントをとることも可能である。

【0059】(セル先頭信号でヘッダを認識) また、ATMスイッチからのセル先頭信号の出力を入力セルバッファの中で保持し、プロセッサによるワード読みだしの時に、当該ワードセル先頭であることをプロセッサ側に通知する。

【0060】(テーブルの検索/対応機能の繰り返しとATMレイヤプロセスの終了) このように整列した状態で入力セルバッファ505に格納されたデータを、受信側ATMレイヤプロセス711が順番に読み出す。プロセッサは、セルヘッダを読み出すと同時に、予め設定されたセルヘッダ値-機能変換テーブル712を検索し、上記テーブル712のセルヘッダ値対応に記述された機能を実行する。ヘッダが有効であることは、セルがATMスイッチを通過する前に受信側の物理レイヤで確認済みなので、ここで強いてヘッダのチェックは行う必要はない。ただし、スイッチ機能検査などのために、ヘッダやルーティングタグを検査してもよい。ATMレイヤプロセス711は、繰り返しセルヘッダを読みだし、対応する機能を実行する動作を予め定められた回数だけ行って、プロセスの実行を終了する。そして、スイッチから入力セルバッファ505にセルが入力され、ある定められた数以上のセルが蓄積すると、再びATMレイヤプロ

セス711が起動される。

【0061】(空セルの処理) さて、例えばセルヘッダが空セルを示す値VPI=VCI=0であれば、セルヘッダ値-機能変換テーブル712の対応するフィールドに記述された動作すなわちヘッダに続くペイロード部分(48byte)のデータの廃棄をFIFOに支持する動作を行う。データの廃棄とは、データをプロセッサのバスに送出することなく、次に読み出されるべき入力セルバッファの先頭の内容を消去することである。データを廃棄することによって、不要なデータがバスを占有する無駄を減らすことができる。ATMレイヤ以上のレイヤでは、空セルのペイロードを参照することはないので、ペイロードを消去することはいくら問題ない。空セルについては、プロセッサが読み出したヘッダの廃棄する。

【0062】(コネクション対応機能の記述法) コネクション対応機能テーブルに記述する方法は、例えば空セル廃棄機能を実行するサブルーチンの先頭アドレスを記述する方法などがある。この場合には、受信セルのVPI/VCI値=0に合致するテーブルのエントリに対応するフィールドのアドレスに制御を移すことにより、空セル機能721が実行される。

【0063】(無効セルの扱いと判定法) コネクション設定されていないユーザセル、OAMセルで、OAM機能が扱わない無効セルなどについては、空セルと同じようにペイロードを読むことなく廃棄しても良い。例えばコネクション設定されていないVPI/VCI値のセルは、ペイロードを読むことなく廃棄する。メタシグナリングチャンネルを扱わないインタフェース装置でのメタシグナリングチャンネルを示すヘッダ値のセルなども同様に廃棄する。

【0064】テーブルには、設定されたコネクション、OAM機能が扱うタイプのセルヘッダ値のみを記述しておき、それらに含まれないものを無効セルとして扱う。空セルもテーブルに記述することを省き、無効セルと同様に扱っても良いが、セル統計情報について空セルと無効セルの扱いは異なるので、セル単位の統計情報が必要な場合には別とする。

【0065】(テーブルの検索方法) セルがOAMセルかユーザセルかを判定するには、VPI/VCI値とPTIの両方のフィールドを見る必要がある。このような場合、それぞれのフィールドだけを検査するために、ヘッダの特定のビットにAND演算を行い、複数のフィールドに対して行ったそれらの判断を組み合わせて判定を行なうことが一般的である。

【0066】この方法によって、単純にセルヘッダの値を比較するやり方に比べて、ヘッダ検査に必要なステップ数やテーブル量を削減することができる。

【0067】(検索方法の動的変更) さて、テーブルの検索には良く知られたハッシュ、番兵等のテーブル検索

10

20

30

40

50

高速化の技法が応用できる。

【0068】その他に、テーブル検索を高速化するために、入力セルの発生頻度に応じて検索テーブルの配列を変更する手法がある。だが、従来はヘッダ検索順序の変更はコネクション単位に限られており、コネクションの記述されたテーブルの配列を変更するのみだった。

【0069】ヘッダテーブルの検索時間は検索手順はもちろん、テーブルの設定コネクション数やコネクション種別毎のセルの発生頻度にも依存する。また、検索時間最短となる検索手順もこれらのパラメータによって異なるが、検索手順を動的に変更することにより、テーブル検索時間の最適化を図ることができる。

【0070】ところが、従来の方法ではコネクションのセル発生頻度については検索効率を最適化することはできても、ユーザセルとOAMセルの間での頻度変化に対して検索効率を最適化することはできない。なぜならば、OAMセルを判定するための特定のビットフィールドを見る操作はコネクション設定テーブルの外、すなわちプログラムのコード内に書き込まれてしまっていたためである。

【0071】ここではヘッダ検査手順を固定せず、セルヘッダを比較するためのヘッダテーブルを検索手順を記述した書換え可能なテーブルとすることによって、動的に検査手順を変えることも可能とする。検索手順を記述したテーブルは、具体的にはコネクション単位の機能の記述と同様に、ヘッダのコネクション識別子の値や、特定のビットフィールドの値を検査するような検索手順を実行するサブルーチンのエントリのアドレスを記述すれば良い。

【0072】検索の実行はテーブルに記述されたエントリ、例えばユーザコネクション振り分けと対応機能の実行、ローカルに終端するOAMセルの分離とOAMプロセスへの引き渡し等のサブルーチンの実行を、いずれかのサブルーチンがあらかじめ定められた検索終了の戻り値を返すまで順番に繰り返すことによって行われる。

【0073】そして、IWFのATMレイヤ機能とは非同期的に実行されるOAM機能が、一定時間毎に受信セルのヘッダ値毎の到着頻度表に基づいて検索時間が最短になるように検索の順序を記述したテーブルを書き換える。

【0074】例えば通常時、始めにVPI/VC I値についてヘッダの検索を行い、次にPTに基づいてOAMセルとユーザセルの分離を行っていたものとする。ユーザセルが多い場合には、まずVPI/VC I値によってコネクション毎にセル分離する操作をする方が効率がよい。一方、F5(VC Iレベル)セグメントOAMセルはPT Iのみで判別可能である(PT I=100)。故障などの何らかの原因によってF5レベルOAMセルの割合が増えた場合には、まずPT IによってOAMセルを分離する方が検索の効率がよい。また、OAMセルの

終端処理はすべてのコネクションについて共通のレイヤ管理エンティティが行うことも多いので、OAMセルについてはコネクション毎に分離して処理する必要がある場合もある。

【0075】逆に、AALサブレイヤでCOM/EOMのようにセルヘッダによって処理が異なる場合には、ATMレイヤのヘッダ機能振り分けでそれぞれのヘッダ値に対応する機能サブルーチンを実行しても良い。

【0076】OAMプロセスが上記テーブルを書きかえている間にも、随時、ATMレイヤのヘッダ検索処理が起動されることがある。ATMレイヤ処理が書き換えが未完了のテーブルを参照すると、例えば同じエントリを2回検索してしまうなど、正常な検索ができない。この問題は、例えばテーブルを2組持って一つを書きかえ用に、もう一つをATMレイヤの参照用として、書換えが完全に終了すると書き換えたテーブルをATMレイヤ処理が参照するように指示することで解決可能である。

【0077】また、この書き換え操作は必ずしも検索時間最短だけを目的とせず、IWFの内部状態や上位制御装置の判断による指示によって変更しても良い。

【0078】例えば、通常動作では最初にVPI/VC I値についてテーブルを検索して振り分ける検索サブルーチンを実行し、次にPT/CLPを参照して必要な処理をしているIWFで、IWFのプロセッサが輻輳状態に陥った時には受信セルのCLP(Cell Loss Priority)を最初に検索し、CLP=1のセルはどのコネクションのセルであろうと直ちに廃棄する措置をとることによって、CLP=1のセルの受信処理と、対応するテーブルの検索処理の負荷を低減してIWFを輻輳状態から回復させることができる。

【0079】このように検索手順を動的に変更することによって、従来よりも広い範囲で検索と処理の効率が最適化されることが期待できる。

【0080】(OAMセルの受信処理)次に、OAMセルの受信処理について説明する。該セルのヘッダ値がOAMセルを示すものであれば、OAMセル処理サブルーチン722が呼び出される。OAMセル処理サブルーチン722は、OAMプロセス731とOAMセルバッファ732を共有しており、現在の書き込みアドレスと読みだしアドレスを示すポインタを持つ。

【0081】OAMセル処理サブルーチンとOAMプロセスは、別々のプロセスとして実行される。両者の間でバッファの競合制御は、例えば次のような方法で実現することができる。

【0082】(リングバッファによるOAMセルバッファの競合制御)OAMセルバッファ732はリングバッファであり、OAMセルが到着すると、ヘッダ値とペイロードの内容が書き込みアドレスから始まるバッファ領域に書き込まれ、書き込みアドレスポインタは1セルバッファ分増加される。一方、OAMプロセスは、1つの

セルの読み出しが終了すると、現在読んだ読みだしアドレスのポインタを1アドレス分増やす。いずれのポインタも、バッファの終了アドレスまで到達する前に先頭アドレスに戻される。OAMセル処理サブルーチンは、書き込み前に書き込みポインタが読みだしポインタと同じ値でないことを検査するので、両プロセスのアクセスが競合することはない。書き込みポインタが読みだしポインタと同じアドレスを指した場合は、バッファのオーバーフローなので、書き込みを予定していたセルデータを廃棄する。

【0083】OAMプロセスは、バッファに蓄積した全てのセルのセルの処理を終えるまで繰り返し、OAMセルの処理を実行する。ただし、ATMレイヤプロセスの実行優先度はOAMプロセスより高くしておき、ATMレイヤプロセスの実行が妨げられて受信セルバッファがオーバーフローすることを防ぐ。従って、OAMプロセスが処理を実行中にもATMレイヤプロセスが起動して、OAMセルバッファに新たなセルが書き込まれることがある。

【0084】(OAMプロセスの起動条件) OAMプロセスが全てのセルの処理を終えた時、すなわち読みだしポインタが書き込みポインタに追い付いた時、OAMプロセスは休眠する。ATMレイヤプロセスは、実行終了の度にOAMプロセスを休眠状態から起床させることにより、一度休眠したOAMプロセスの実行を再開させる。

【0085】OAMプロセスは、ATMレイヤプロセスに次ぐ優先度で実行される。

【0086】(ポインタ渡しのメッセージを使う方法) ここまでに説明したリングバッファとポインタを使うバッファ制御方式の他に、ATMレイヤプロセス(OAMセル処理サブルーチン)とOAMプロセスとの間で、書き込みバッファアドレス及び処理済みバッファアドレスのポインタをメッセージとして送受信して、バッファの排他制御を行う方式をとってもよい。ポインタをメッセージとしてやりとりする場合の詳細は、以下に述べるAAL機能と同様なので省略する。この場合は、メッセージの受信がOAMプロセスの起動条件になる。

【0087】(AAL機能) 次に、ユーザコネクションのAAL機能サブルーチンについて説明する。

【0088】OAMセルや空セルと同様に、ヘッダ値がヘッダ機能変換テーブル712に記述された値に適合するAALコネクションのセルが到着すると、対応するフィールドに記述されたサブルーチンが実行される。ここでは、AALサブルーチン723を指定するセルが到着したものとする。

【0089】(AALコネクション対応のバッファ管理) AAL5(FR-SSCS)コネクションでは、到着セルヘッダのPTI値によって処理が異なる。AAL5のバケットは、末尾のセルがATM-UU=EOMで、

残りのセルはATM-UU=COMのセルで構成される。

【0090】(COMのセルの扱いバッファ構成) はじめに、ATM-UU=COMのセルが到着した時の動作を説明する。ATM機能から実行されたAAL機能サブルーチン723は、セルペイロードをAAL5用に定義されたバッファに格納していく。AAL機能は、少なくともそのコネクションで予め定められた最大バケット長のデータを複数個格納できる連続したバッファ領域AAL5バッファ742を持つ。図3においては、3つのバケットを格納できるバッファ構成を示し、バッファ1~3、751~753がそれぞれ最大バケット長を格納できる。

【0091】(COMセルの扱いバッファオーバーフロー) AAL機能サブルーチンは、バッファの現在の書き込みアドレスを保持する書き込みポインタを持っており、入力セルバッファ505から読みだしたセルペイロードを該ポインタの示すアドレスに書き込みながらポインタを更新していく。もし、バッファ長がコネクションで予め定められた最大長を越えて、バッファの最終アドレスを越えてしまう場合は、それまでに蓄積されたデータは無効として廃棄され、書き込みポインタその他の変数と後述のCRC計算機能の値は予め定められた初期値に戻される。AAL5では、この最大長はMax_SDU_Deliver_Lengthとしてコネクション設定時に指定される。輻輳などによってAAL5コネクションのEOMセルが廃棄された時には、このような現象が起きる場合がある。

【0092】(CRCの計算) 受信セルのペイロードは、前記バッファ書き込みポインタの指すアドレスに書き込まれる。このとき、同時にCRC32の計算も行われる。CRC32の計算をソフトウェア的に行うにはn bitのデータに対してn回のシフトと平均でn/2回の(データによって変わる)排他的論理和の計算が必要になる。これは、データ転送の時間に比べてかなり大きい。例えばFIFO及びメモリへの転送のペナルティをそれぞれプロセッサのシステムクロックの4倍と仮定すると、1ワード(32bit)の転送時間は8クロックとなる。だが、1ワードのCRCを計算するには、少なくとも $32 \times 3 / 2$ クロックの時間が必要になる。CRC計算を高速化することで、メモリ転送の最適化以上にAAL5機能のプロセッサ処理を高速化することができる。

【0093】ここでは、補助的な専用ハードウェアCRC計算機能508を用いて、プロセッサによるメモリへの書き込みと同時にそのハードウェアがそのデータを取り込み、CRC計算を行う。CRC計算の回路は、シフトレジスタを用いたものが広く知られている。

【0094】まず初期状態では、CRC計算回路のレジスタ値は0に設定される。計算回路は、プロセッサのメ

メモリ上のAALバッファ領域742に値が書き込まれた時に、次の計算を行う。その他のメモリ領域、例えばプログラムやスタックの領域をプロセッサがアクセスしても、CRC計算機能は無視する。AAL5での誤り検査符号はCRC32と呼ばれ、 $M(X)$ を情報例、 $G(X)$ を生成多項式とする一般のCRC即ち $M(X)X^k \bmod G(X)$ の値をそのままは用いない。CRC32では、 $(M(X)X^{32} + (X^{31} + \dots + X + 1)X^k) \bmod G(X)$

$$\text{result} = (RX^N + D) \bmod G(X)$$

R レジスタに保持されている値
D メモリに書き込まれるデータ
N バスの幅 (bit)
G(X) 生成多項式

プロセッサからメモリへの書き込みアドレス出力が予め定められたバッファの範囲である時に該CRC計算回路に値が書き込まれてCRC計算が行われるような回路を構成することは容易である。ただし、データバスの幅Nは生成多項式G(X)の次数の整数倍でなければならない。G(X)の次数は32でセルも48byteすなわち4byte(32bit)の倍数なので、N=32の時にはバケットを構成するセルの転送終了とともにCRC計算回路にCRC計算の結果が格納される。CRC32の計算のためには、バケット先頭セルの最初の4byte(32bit)を反転したデータについてCRCの計算を行えば良い。なお、ここではデータバスが32bitの場合について述べたが、データバス幅が64bitになってもまだこの条件が成立する。

【0096】(CRC計算の速度について)CRC計算回路がプロセッサからメモリへのデータ書き込みサイクル内に終了する程十分高速に動作すれば、プロセッサが連続的にメモリにデータを書き込んだとしても正しくCRC計算ができる。

【0097】一方、CRCがデータ書き込み1サイクルのうちに終了しない場合には、CRC計算が継続中であることを検出する回路を該回路に設け、計算が継続している間は書き込みを待ち状態に保留することによって正しいCRCを求めることが可能である。

【0098】(COM/EOM時の処理CPCS-LP/C Iの求め方)受信セルのヘッダのCLPまたはC Iのビットが1になっているセルがあれば、当該AAL機能がローカルに持つ変数CPCS-LPまたはCPCS-C Iを1に設定する。これらの変数は、ITU-T勧告I.363.6に記述されるCPCSサブレイヤの同名のプリミティブCPCS-LP、CPCS-C Iに対応し、初期状態では値0が代入される。

【0099】以上の処理を行ってAAL機能サブルーチンは終了し、実行はATMレイヤに復帰する。当該コネ

* $d G(X) \pmod{2}$ の和の1の補数を取ったものが用いられる。これは、一般のG(X)を生成多項式とするCRCの結果に一定値を加えたものと同値である。以下では説明を簡単にするため、一般のCRCの計算方法に基づいて説明する。

【0095】

【数1】

クションのセルの到着の度にサブルーチンが実行され、バッファにバケットが蓄積される。

【0100】(EOMの処理)受信セルのATMユーザ情報がバケットの終りを表すEOMの場合は、パイロード転送が完了してCRCの計算完了後、計算回路の値を期待された値と比較する。比較の結果が不一致ならばビットエラーまたはセル廃棄が発生しているため、当該バケットはエラー(無効)であり、AAL機能サブルーチン723の変数を初期化する。

【0101】EOM受信時にはデータが有効/無効いずれの場合にも、データ書き込みポインタ、受信セル数、CPCS-LP、CPCS-C Iの各変数とCRC計算機能の内部レジスタは予め定められた初期状態へ戻される。

【0102】(ATMレイヤプロセス(AALサブルーチン)→ATM→FRプロセスへのデータ渡し)有効なバケットが検出された場合、AALサブルーチンはバッファ751の先頭アドレスとバケット長、及びCPCS-LP/C Iの値をATM-FR変換プロセス741に通知する。ATM-FR変換プロセスは、この通知によって起動される。具体的には、この通知は、OSのメッセージボックス機能が用いられる。

【0103】(EOM終了処理)次に、AALサブルーチンは、ATM-FRプロセスが処理終了したバッファ752のアドレスをメッセージボックスから取り出し、書き込みポインタのアドレスとした上で、その他の変数を初期化してサブルーチンを終了する。もし、メッセージボックスが空であればバッファ領域が不足しているので、例えば書き込みポインタに値0などのバッファアドレスが無効であることを示す値を代入するなどして、次に到着するセルの格納を中止する。

【0104】(バーストとバッファ量の関係)ATM側の入力バースト的に発生してフレームリレー側の出力が間に合わない場合に、このような現象が生じる。大き

なバーストを許容するには、AALバッファ742に含まれるバッファの個数を増やす必要がある。セルの到着頻度が常に一定ならば、バッファの個数はAAL組み立て用とフレームリレー出力用の2個だけで良い。

【0105】(バッファオーバーフロー対策)書き込みポインタが無効な状態においても当該AALコネクションのセルが到着する度にAALサブルーチンは起動され、データ廃棄動作を行う。もし、メッセージボックスにATM-FR変換プロセスが処理を終了したバッファのアドレスが通知されていれば、そのアドレスを書き込みポインタとして初期設定を行い、セルの格納を開始する。

【0106】(ATM-FR変換プロセス)次に、ATM-FR変換プロセスについて説明する。

【0107】(フォーマット変換)さて、次にデータの処理を行うのは、ATM-FRプロセスである。ATM-FR変換プロセス741は、AAL機能サブルーチン723からパケットに関する情報をメッセージとして受信することによって起動される。

【0108】(制御メッセージの分離)AALバッファに再構成されたパケットは、トレイラが末尾に附属したCPCS-PDU921の形式である。このヘッダ部分922に必要な変更を加えてFRのレイヤ2(以下L2とする)パケット形式に変換し、ヘッダ部分922と情報フィールド923をFR-PHYに転送するのがATM-FR変換プロセスである。

【0109】変換プロセスは、まずヘッダに含まれるデータリンク識別子を認識する。データリンク識別子が受信コネクションのものであり、パケットが制御/保守メッセージでないことを確認してパラメータ設定処理を開始する。もしパケットが制御/保守メッセージであれば、OAMプロセスに通知して所定の処理を行う。

【0110】(ヘッダの書き換え)FRのL2パケット形式への変換は、パケットのヘッダのうちの幾つかのフィールドの値を書き換えるだけである。

【0111】CPCS-LPの値すなわち受信コネクションのCPCS-LP値とパケットに含まれるデータ優先度(Data Eligibility)の値のORをとったものを、パケットのデータ優先度(Data Eligibility)のフィールドに書き込む。

【0112】FECNには、元のパケットのFECNフィールド自身の値とCPCS-CIの値のORをとったものをパケットのFECのフィールドに書き込む。

【0113】(DLCI書き換え)もし物理リンク毎にデータリンク識別子DLCI書き換える必要がある場合には、入り側DLCIと出側DLCIを対応した表に基づいて入り側DLCIを鍵として出側DLCIを検索し、DLCIを書き換える。

【0114】(FR-PHYへのデータ出力)これで、パケットフォーマットの変換は終了する。このパケット

のヘッダ及び情報フィールドを、L2パケットの送信処理を行うハードウェアFR-PHY507に転送する。

【0115】送信処理は、具体的にはフレーム前後のフラグパターン、FCS(CRC16)の付加、0挿入と物理レイヤ処理(ラインコーディング)である。

【0116】(転送時間の問題)本実施例では、ATMセルのSAR処理を行うためのバッファをFR-PHYへのデータ出力用バッファと共用しているために、FR-PHYへデータ転送を行っている間はATMレイヤプロセスが該バッファをアクセスすることができない。FR-PHYへのデータ転送の間に入力セルバッファのオーバーフローが生じないように、FIFOの容量を十分大きくとっておくことが必要である。

【0117】(転送手順)まず、プロセッサがFR-PHYに新しいパケットの出力を開始する指示を発行し、予め各AALコネクション対応に用意されたパケットの先頭を示すアドレスから始まるデータをプロセッサが読み出し、前記FR-PHYに設けられたデータ出力用レジスタに書き込んでいく。全てのデータ書き込みが終了すると、パケットのデータ長を含むデータ出力終了指示をFR-PHYに発行する。

【0118】この処理は、例えばプロセッサがFR-PHYに付随するDMA機能に送信開始を指示し、プロセッサメモリの内容をDMA機能によって読みだしてFR-PHYのデータ処理/送信機能へ転送して処理するという実装形態をとっても良い。

【0119】ただし、ハードウェア機能を簡素化するという観点からは、DMAのようなバス調停を伴う複雑な機能は避けることが望ましい。今までに説明した通信プロトコル処理においては、メモリアクセスが大きな比重を占めるため、DMAがバスを占有している状態では処理が進行できないためである。ただし、FR-PHYとプロセッサ間のバスインタフェースについてはプロセッサを使った場合には、データ転送効率が直接処理時間を左右するため、高速なものを使うことが望ましい。

【0120】バス調停機構を簡略化しつつ、データ転送の間にプロセッサバスが占有される問題を回避する手段として、FR-SSCS機能のパケット用バッファのために確保されるメモリを2portのRAMとして、片方のポートをFR-PHYのDMAバスに接続する方法がある。図6に、その構成図を示す。このようにすれば、FR-PHYのデータ転送中にも他の処理を進めることができる。ただし、2port RAMは単一port RAMに比較して複雑であり、コスト高を招いてしまう。

【0121】(FR-PHYにおける処理)FR-PHYでは、入力データの内容についてCRC-16の計算手順に基づいてCRCが計算され、パケットの末端に付加される。そして、フレームの前後にフラグパターンが付加される。このデータ内容にフラグパターンと同じボタン

が出現することを防ぐために、1ビット単位で値0のビットデータが挿入され、このデータに本来の物理レイヤ（レイヤ1）処理すなわちラインコーディングの処理が行われて、フレームリレーインタフェースのライン出力としてデータが送出される。

【0122】（ATM-FR変換プロセスの終了）ATM-FR変換プロセス741は、FR-PHYへのデータ出力を終えると、出力を終えたバッファのアドレスをAALサブルーチン723のメッセージボックスに出力して、該バッファは再びセルの格納のために使われる。10 以上で、ATM-FR変換プロセスは終了する。

【0123】（FRからATMへの変換）次に、FRからATMへの変換機能について図4を参照しながら説明する。

【0124】（物理レイヤ）フレームリレーインタフェースのラインから入力された信号は、受信側FR-PHY506においてITU-T勧告I.431に基づく処理、すなわちライン入力、クロック抽出、ビット再生、デスクランブル、フレーム同期、L2フレーム検出（time fill除去）を施される。PHY-FRは、20 許容される最大長のフレーム2つ分以上のバッファを持ち、プロセッサにパケットの到着を通知してから該パケットの転送が終了するまでの間も、ラインから入力されるデータを受信してバッファへ格納する動作を継続する。また、FR-PHYにおいては、L2フレームの検出と並行して該L2フレームのCRC計算も行う。

【0125】（Q.922フレーム検出）フレームリレーのL2プロトコルデータユニット（以下PDU）は、図5の911に示すフレーム構造をとる。FR-PHYは、フレーム終了側のフラグボタンを検出すると、割り込み信号をプロセッサに入力してフレームの到着を通知する。30

【0126】（FR-ATM変換プロセスの起動）割り込みを受けたプロセッサは、FR-ATMプロセス841を起動する。受信側FR-ATMプロセス841は、FR-PHY506が受信データについてCRCを計算した結果を期待される値と比較し、両者が異ればエラーとしてパケットの廃棄をPHY-FRに指示する。この結果は、プロセッサのメインメモリ上にマッピングされた予め定められたFR-PHYのレジスタに表示される。40

【0127】（ヘッダの読みだし前準備）FR-ATMへの変換のために、プロセッサは、L2PDUのヘッダが含まれる1セルベイロード分のデータをFR-PHYから読み出して、以下の処理を行う。

【0128】（パケット格納バッファの確保）予め定められたFR-PHYのレジスタに該パケットのデータ長が表示され、プロセッサはこれを読み出してFR-PHYから転送を行うデータのバイト数を知る。もしデータ長がプロトコル規約上違反しているものであれば、該パ

ケットの廃棄をFR-PHYに指示する。データ転送先のバッファは、ATM-FRへの変換の場合と同様に、当該コネクションで許される最大パケット長のデータを複数個格納できる連続したバッファ領域AALバッファ842を持つ。図4においては、3つのパケットを格納できるバッファ構成を示し、バッファ1～3、851～3がそれぞれ最大パケット長を格納できる。

【0129】ここで、バッファ長をQ、922プロトコルの最大長だけでなく、最大長のパケットをAAL5形式に変換するのに十分なだけの大きさをとっておけば、プロトコル変換に当たってデータの移動やつなぎ合わせを行う必要がなくなる。

【0130】また、AALサブルーチンでの取り扱いを簡単にするためにバッファはセルベイロードの整数倍即ち48の整数倍の長さとする事が望ましい。

【0131】（ヘッダの認識：制御メッセージの分離）FR-ATM変換プロセス841は、データリンク識別子を認識する。データリンク識別子が受信コネクションのものであり、パケットがコントロール/保守メッセージでないことを確認してパラメータ設定処理を開始する。データリンク識別子がコントロール/保守メッセージを示すものであれば、当該パケットの処理をOAMプロセス831に渡してFR-ATMプロセスを終了する。

【0132】（ヘッダパラメータの書き換え）対向側のATMからFRへの変換と同様に、ヘッダパラメータのほとんどはそのままの値で利用される。変更されるのは、FECN、BECN（Forward/Backward Explicit Congestion Notification）とDLCIのみである。

【0133】（DLCI書き換え）もし物理リンク毎にデータリンク識別子DLCI書き換える必要がある場合には、入り側DLCIと出側DLCIを対応した表に基づいて入り側DLCIを鍵として出側DLCIを検索し、DLCIを書き換える。

【0134】（FECN、BECNの書き換え）FECNフィールドには値0を、BECNフィールドには対向側AAL受信機能において受信されたAAL機能623が持つ変数CPCS-CIの値を、それぞれ代入する。これで、パケットの変換は終了する。

【0135】（FR-SSCS機能：PAD）次に、パケットはAAL5CPCS形式921に変換され、CPCSTレイラが付加される。トレイラは、924に示すようなフォーマットからなる。パケットをセル単位に分割する時に、トレイラがセル間に跨って格納されることを防ぎトレイラが常に最終セルの同じ部分に位置してトレイラの認識を容易にするため、トレイラと情報フィールドとの間にはPADと呼ばれる可変長の領域が付加される。PAD領域は、値0のオクテッドで埋められる。

【0136】（FR-SSCS機能：trailer）

CPCSトレイラ本体には、CPCS-UU、CPI、Length、CRCのフィールドがある。CPCS-UUは、FR-SSCSでは値0に設定される。CPIの利用法はまだ未定であるが、当面值0で埋めることが規定されている。Lengthフィールドは、ペイロードのオクテット数を入れる。CRCフィールドには、CRC32の計算結果が格納される。これで、AAL5-CPCS形式のバケットが準備できたことになる。このバケットの先頭アドレスとセル数をFR-SSCSコネクションに対応するAALサブルーチン823の入力キューに渡し、FR-ATM変換プロセスは終了する。この通知は、具体的にはOSのメッセージボックス機能等が用いられる。

【0137】(AAL機能へのデータ渡し) 出力側AAL機能823は、出力側ATMレイヤプロセス601から呼び出される。AAL機能は、初期状態ではまず、FR-ATM変換プロセス841からの入力キューから出力バケットの先頭アドレスとセル数を読み出して、出力セル表示824をセルありに書き換える。

【0138】キューが空の場合は、出力セル表示をセル無しとして終了する。

【0139】初期化が終了している場合、まず転送セル数を表す状態変数を1増加し、FR-ATM変換プロセス841から渡された転送セル数と比較し、転送終了ならば予めコネクションデータが書き込まれたヘッダデータのPTを、EOMそうでなければPTにCOMを書き込んで、ATM-SWバッファに出力する。そして、先頭アドレスの示すメモリから1セルペイロード分の48 byteのデータを読み出して、ATM-SWバッファに書き込む。転送を終えると、先頭アドレスを48 byte増加して、その値を保持しておく。

【0140】転送終了ならば、転送アドレスと転送セル数の2つの変数を初期状態に戻して、FR-ATM変換プロセスからの入力キューから出力バケットの先頭アドレスとセル数を読み出して、変数を初期化する。キューにデータがない場合は、出力セル表示をセル無しとして終了する。

【0141】バケットの転送が途中の場合も、1セルデータの転送完了とともにAALサブルーチンは終了して、制御をATMレイヤプロセスに戻す。

【0142】(送信側ATMレイヤプロセス) 送信側ATMレイヤプロセスは、各ユーザコネクション毎にコネクション設定時に合意したトラヒックパラメータを守るように、予め定められたアルゴリズムにしたがって順番にセルの出力権をOAMあるいはユーザのコネクションに割り当てる。設定されているコネクションとトラヒックパラメータは、ヘッダ機能変換テーブル812に記述されており、ATMレイヤプロセスはこの表と各コネクションの出力セル表示822、824に基づいてセルを出力するスケジュールを決定する。

【0143】先に説明した通り、呼び出された各コネクションセル出力サブルーチンは、一度の呼出毎に1個のセルを出力する。

【0144】スケジューリングの際にセルを出力すべきコネクションを割り当てられない時は、空セル出力を割り当てる。空セル出力は、空セル出力専用のサブルーチンを持っても良いが、空セルは一定パターンなので、空セルバタンの出力機能をATM-SWバッファに組み込み、プロセッサが空セル出力の指示を行うようにしても良い。

【0145】(ATMレイヤプロセスの優先度) ATMレイヤプロセスは、起動時に一度各コネクション対応の出力セル表示を読み、セル出力のスケジュールを決定する。従って、出力セルのないコネクションのセル出力サブルーチンが呼び出されることはない。ATMレイヤプロセスは一連のFR-ATM変換に関わるプロセスの中でもっとも優先度が高いため、ATMレイヤプロセスの実行中に各コネクションの出力セル状態が変化してしまうことは考慮しなくても良い。

【0146】出力セルの有無に関わらず、テーブル812に基づいて予めスケジュールを定めておき、各コネクションのセル出力サブルーチンを呼び出しても良いが、出力セルのないコネクションのサブルーチンが何度も呼び出される無駄がある。

【0147】このようにしてATMとフレームリレーを相互変換するIWFを実現できる。

【0148】[複数インタフェース]これまでの説明ではAALコネクション及びフレームリレーインタフェースが1つだけの例を示したが、IWFが複数のフレームリレーインタフェースを持ち、それに複数のATMコネクションが接続される場合にも既に説明したハードウェア/ソフトウェア構造が適用できる。複数コネクションに対応するには、AALのバッファ742、842およびコネクションのバッファ管理用変数(書き込みポイントなど)とFR-PHY506をコネクション数だけ増やすことにより容易に実現可能である。CRC計算機能については、1セルペイロードの転送毎にCRC計算機能508の内部状態をプロセッサメモリに退避することで複数コネクションに対応できる。また、ATM-FR(図3)、FR-ATM(図4)ごとにCRC計算機能を設けることなく、計算機能を共有してもよい。

【0149】本実施例では、プロセッサの周辺装置からプロセッサのメモリ上で一度バケットを再構成し、バケットフォーマットの変換を行って周辺装置に出力している。プロセッサバス占有率の削減=データの転送回数の削減という観点から、プロセッサメモリの上でバケットを再構成することなく、周辺装置から周辺装置へ直接データを転送するような構成も実現可能である。

【0150】以上、本実施例の基本構成を説明してきた。

【0151】以下では、本実施例のCRC計算機能について詳細に説明する。

【0152】＜AAL5とEther間のプロトコル変換装置の一例＞まず、CRCの方式が同一であるAAL5とEtherとの間のプロトコル変換におけるCRCの計算省略方法について説明する。

【0153】図7は、ATM、AAL5、Ethernet（IEEE802.3）のケットフォーマットを示したものである。

【0154】図7において、1101はイーサネットの10ケット、1111はケットのヘッダ、1115はケットの情報部、1116はケットのFCS（CRC32）をそれぞれ表す。

【0155】また、ケットのヘッダは、送信先番地1112、発信元番地1113、長さ情報1114から構成される。

【0156】1102は、AAL5のCPCS-PDUであり、ヘッダ1121、情報部1122、PAD1123、CPCS-UU1124、CPI1125、length1126、CRC32（1127）のそれぞれ20で構成される。

【0157】1103-1から1103-nは、CPCS-PDUの分解または組立で生成または受信されたATMセルである。

【0158】図8は、AAL5とEtherとの間のプロトコル変換を行う場合の装置構成の一例を示したものである。

【0159】図8において、501はプロセッサ、502はプロセッサバス、503はバッファメモリ、504は出力セルバッファ、505は入力セルバッファ、1206はイーサネットインタフェース（Ether-I/F）、508はCRC計算機能、508はATMスイッチである。

【0160】（フレームリレーとイーサネットの相違点）前述した本実施例の基本構成においては、フレームリレーとATMを相互接続するIWFを説明したが、同*

*様にATMとイーサネット（IEEE802.3）を相互接続するIWFを考えることができる。このIWFのインタフェース仕様は、例えばATM forum LAN Emulation Over ATMなどがあるが、ここでは単純にAAL5のSSCS（サービス依存部分）がNULLで、そこにイーサネットのケットを乗せておくこととする。

【0161】図8のように、ATMとイーサネットの相互接続も、フレームリレーのFR-PHYをイーサネットの伝送処理を行うEther-I/F1206に置き換えるだけでよい。処理方法についても、ケットフォーマットの違いをヘッダ書き換えに反映する程度で、前述した本実施例の基本構成から容易に類推可能である。ここでは、前述のフレームリレーの場合との違いについてのみ説明する。

【0162】まず、イーサネットではQOSが定義されていないため、ケットがある時間内に必ず出力できるとは限らない。フレームの衝突が発生している間はフレームの送出ができないのである。フレームの送出ができない時に新たなフレームの到着が続けば、フレームを廃棄せざるを得ない。

【0163】この問題を解決するには、ATMコネクションの側でフロー制御を行う方法がある。または、ATMコネクションをQOSなしのコネクションとしてフレーム廃棄を許容することを前提として利用しても良い。

【0164】フレームリレーでは、ケットの誤り検出符号はCRC16で、AAL5のCRC32とは計算方法が違う。だが、イーサネットのケットの誤り検出符号はAAL5のCRC32と同じなので、AAL5とイーサネットのフレームのフォーマット変換に対する不変部分のCRC32の計算を省略することができる。以下この発明について説明する。

【0165】イーサネットのケット1101の誤り検出符号FCSは、次のように表される。

【0166】

【数2】

$$\begin{aligned} C(X) \bmod G(X) &= (A(X) X^{L_B} + B(X)) \bmod G(X) \\ &= (A(X) X^{L_B}) \bmod G(X) + B(X) \bmod G(X) \\ &\dots (1) \end{aligned}$$

A(X) ケットのヘッダ部分1112, 1113, 1114の多項式表示
B(X) ケットの情報部分1115の多項式表示
C(X) ケット1101全体の多項式表示
 L_B ケットの情報部分1115のビット数
G(X) CRC32の生成多項式

一方、CPCS-PDU1102の誤り検出符号は、次のように表される。

【0167】

【数3】

$$\begin{aligned}
 V(X) \bmod G(X) &= (V(X)X^{L_T} + T(X)) \bmod G(X) \\
 &= ((A'(X)X^{L_B} + B(X))X^{L_T} + T(X)) \bmod G(X) \\
 &= (A'(X)X^{(L_B+L_T)} + B(X)X^{L_T} + T(X)) \bmod G(X) \\
 &= \underbrace{(A'(X)X^{(L_B+L_T)} \bmod G(X))}_{\text{第1項}} + \underbrace{(B(X)X^{L_T} \bmod G(X))}_{\text{第2項}} + \underbrace{T(X) \bmod G(X)}_{\text{第3項}} \\
 &\quad \dots (2)
 \end{aligned}$$

$V(X)$ CPCS-PDU1102の多項式表示
 $W(X)$ CPCS-PDUの情報部分の多項式表示
 $A'(X)$ パケット1102のヘッダ1121のフォーマット
 変更された多項式表示
 $B(X)$ パケット1102の情報部分1122の多項式表示
 $T(X)$ CPCS-PDUのCRCを除いたトレイラ部分
 1123, 1124, 1125, 1126の多項式表示
 L_T パケットのトレイラ部分のビット数

(2) 式第2項は次のように変形できる。

*【数4】

【0168】

*

$$(B(X)X^{L_T} \bmod G(X)) = ((B(X) \bmod G(X))X^{L_T} \bmod G(X))$$

... (3)

つまり、(3) 式を求めるためには、(1) 式の結果の第2項を利用できる。

【0169】さて、プロセッサでCRCの計算を行う場合には、被演算数のビット数オーダーの演算が必要なのは前述した本実施例の基本構成で触れた。従って、

(1) の右辺第2項 $(B(X) \bmod G(X))$ から

(3) を得るには、 $3/2 \times L_T$ 回のビットシフト/XOR演算を行えば良い。 L_T はトレイラのビット数で、64~440回の演算が必要であるがパケット全体のビット数に比べればごく少い。

【0170】イーサネットのパケットの最大長1600 octetと最短長36 octetの平均984 octetとトレイラ部分の平均の長さ31 octet (55と8の平均) を比較すると、トレイラ部分の全パケットに占める割合は3%程度である。

【0171】それぞれの処理時間を考えてみる。前述した本実施例の基本構成と同様に、FIFO及びメモリへの転送のペナルティを、それぞれプロセッサのシステムクロックの4倍と仮定する。ペイロード部分の1ワード (32 bit) の転送時間は8クロックとなり、全ペイロードの転送時間は984×8クロック (読み込み+書き込み) で7872クロックである。ハードウェア処理のCRC計算は、これと同時にされる。

【0172】トレイラ部分のCRC計算では、トレイラ長の平均長、31 octetのCRC計算はメモリアク※

※セスが4 (word) × 4クロック (読み込みのみ)、シフト/XOR演算が248×3/2で合計388クロックかかる。

【0173】ペイロード部に対するトレイラ部の処理時間の割合は、388/7872=0.049で約5%である。

30 【0174】トレイラは必ずしも4バイトでアライメントがとられているわけではないので、この計算では4バイトのワード単位で動作するCRC計算回路が使えるとは限らない。

【0175】トレイラ部の演算を専用ハードウェアで行うには、オクテット単位すなわちシフト演算の回数を8回単位を任意に設定できるようなCRC計算回路が必要である。だが、これまで述べたように L_T はあまり大きくないので、計算はプロセッサによる処理で行っても全体の性能は5%悪化する程度である。

40 【0176】また、予めオクテット単位に計算したCRCの結果を格納したテーブルを組み合わせるソフトウェアによるCRC計算処理を単純なビットシフト処理に比べて、数倍高速に処理することが考えられるが、トレイラ部分の計算にこの方法を利用することも可能である。

【0177】(2) 式の第1項

【0178】

【数5】

$$(A'(X)X^{(L_B+L_T)} \bmod G(X))$$

... (4)

の計算では $A'(X)$ は $A(X)$ から書換えられているので、(1)の結果を使ってシフト/XOR演算の回数を減らすことはできない。従って、データのビット数 $(L_A + L_B + L_T)$ オーダーの演算が必要になる。だが、(4)でデータの可変領域は $A'(X)$ の先頭の14 byteだけで、残りは $(L_B + L_T)$ bitの値0の固定値である。

【0179】演算の開始時に $A'(X)$ のデータと L_B の数を与えて(4)の結果を計算する回路は、CRC計算回路に L_T のだけ値0のビットを入力する回路を付加することにより実現できる。この計算には当然 $L_A + L_T$ 回の演算が必要であり、 $L_A + L_T \gg 14 \times 8$ (ヘッダ部 $A'(X)$ のビット長)ならば、 $V(X)$ の計算とほぼ同じオーダーの時間がかかるが、プロセッサから見*

$$T(X) \bmod G(X)$$

はCRC計算機能またはソフトウェア処理によって計算する。この計算の繰り返しオーダーは、ビット数32～408回であり、さほど多くはない。

【0183】これで(3)、(4)、(5)の値の和をとることによって、(2)式の値が求められる。

【0184】実際の処理手順を図9に従って説明する。*

$$C(X) = A(X) X^{L_B} + B(X)$$

また、AAL5パケット全体1102、ヘッダ部1121、情報部1123、トレイラ部1123～1127をそれぞれ、 $C'(X)$ 、 $A'(X)$ 、 $B'(X)$ 、 T' ★

$$C'(X) = A'(X) X^{(L_B + L_T)} + B'(X) X^{L_T} + T(X) \quad \dots (7)$$

まず、処理ステップS1311、S1312で、ヘッダ部分1112～1114、情報部分1115の転送と同時にそれらのCRC1301、CRC1302が計算される。ステップS1313でイーサネットパケット1101全体のCRCがそれらの和(1303)として求められる。

【0188】AAL5形式パケットのCRC計算は、次のように行う。

【0189】まず、ステップS1314でパケット先頭の書き換えと同時にそのCRCを計算する。既に述べたように、CRC1302の結果を用いることにより、その計算量はトレイラのビット数分 $(X \text{ の } L_T \text{ 乗})$ だけで済む。ステップS1316でトレイラの付加と同時にトレイラ部分1123～1126のCRCを計算し、ステップS1317でそれらの和をとることによって、AAL5形式のパケット1102の全体が計算できる。

【0190】<フレームリレーとATMの間のプロトコル変換装置の一例>上記のAAL5とEtherの間のプロトコル変換装置の一例で説明した同一データの同一

＊る限り1回のデータ書き込みで(4)式の値が求められる。従って、計算の所要時間そのものは本実施例の基本構成で示したメモリ書き込みと同時にCRC計算を行う方式と同等のオーダーだが、この間バスアクセスの必要がないためにプロセッサと並行して処理を進めることができ、プロセッサとバスへの処理の負荷が小さくなる。【0180】また、多数のコネクションが多重化されている場合、1セルを受信する度にCRC計算機能の値を退避/回復させなければならないが、このやり方であればCRC計算機能の状態を退避/回復する必要がなくなる。

【0181】(2)式の第3項

【0182】

【数6】

... (5)

※【0185】イーサネットパケット全体1101、ヘッダ部1112～1114、情報部1115をそれぞれ、 $C(X)$ 、 $A(X)$ 、 $B(X)$ とすると、次の式が成立する。

【0186】

【数7】

... (6)

★(X)とすると、次の式が成立する。

【0187】

【数8】

CRC計算を省略する方法は、CRCの生成多項式が異なる場合にも拡張が可能である。

【0191】以下では、CRCの方式が異なるフレームリレーとATMとの間のプロトコル変換におけるCRCの計算省略方法について、前述した本実施例の基本構成のフレームリレーとATMのIWFに該CRCの計算省略方法を適用する場合について説明する。

【0192】(不変部分) 図5において、フレームリレーパケットの情報フィールド913とCPCS-PDUにおける対応部分923は、パケットフォーマット変換の前後で変化しない。それぞれのパケットで計算されるCRCは、生成多項式がそれぞれ16次と32次と異なっていて互いに素である。

【0193】まず、実施例2で説明した方法を用いて、これらの不変領域のCRCを他の部分と独立して計算することができる。

【0194】

【数9】

31

$$A(X) \bmod G_1(X)$$

... (8)

32

$$A(X) \bmod G_2(X)$$

... (9)

A(X) ビット列913(923)の多項式表示

G₁(X) CRC16の生成多項式G₂(X) CRC32の生成多項式

さて、ここで2つの生成多項式G₁(X), G₂(X) *る。

の積G(X)を考える。G(X)の次数は16+32= 【0195】

48である。A(X)をG(X)で割った剰余を求め *10 【数10】

$$A(X) \bmod G(X) \quad \dots (10)$$

G₁(X), G₂(X)は互いに素なので(8), ※【0196】

(9)について次の式が成立する。 ※【数11】

$$(A(X) \bmod G(X)) \bmod G_2(X) = A(X) \bmod G_1(X) = (7)$$

... (11)

$$(A(X) \bmod G(X)) \bmod G_1(X) = A(X) \bmod G_2(X) = (8)$$

... (12)

このように2つの生成多項式の積を生成多項式として剰余を求め、それを本来の生成多項式で割ることによっても、(9), (10)を求めることができる。剰余を本来の生成多項式で割る演算は、G(X)の次数オーバーすなわちこの場合では高々48次なので処理負荷はごく小さい。

【0197】生成多項式の次数が32次から48次となっても、CRC計算機能のハードウェア規模は高々1.5倍になるにすぎない。また、プロセッサによって計算する場合には、64bit CPUを使えば処理ステップ★

20★数は全く変わらない。

【0198】このようにして、生成多項式が異なる場合についても、CRCの計算を共通化することができる。

【0199】実際の処理手順を図10に従って説明する。

【0200】フレームリレーパケット全体912、ヘッダ部912、情報部913をそれぞれ、C(X), A(X), B(X)とすると、次の式が成立する。

【0201】

【数12】

$$C(X) = A(X) X^{L_B} + B(X) \quad \dots (13)$$

また、AAL5パケット全体912、ヘッダ部922、情報部923、トレイラ部924~928をそれぞれ、C'(X), A'(X), B'(X), T'(X)とす☆

☆ると、次の式が成立する。

【0202】

【数13】

$$C'(X) = A'(X) X^{(L_B+L_T)} + B'(X) X^{L_T} + T'(X) \quad \dots (14)$$

G(X) = G₁(X) G₂(X), G₁(X)は、フレームリレーのCRCの生成多項式、G₂(X)はAAL5のCRCの生成多項式とする。

【0203】まず、処理ステップS1411, S1412でヘッダ部分912、情報部分91の転送と同時にそれらのG(X)による剰余1401, 剰余1402が計算される。ステップS1413でフレームリレーパケット921全体の剰余がそれらの和(1403)として求められる。

【0204】AAL5形式パケットのCRC計算は、次のように行う。

【0205】まず、ステップS1415でパケット先頭の書き換えと同時にそのCRCを計算する。そしてステップS1316での情報部分923のCRC計算では、

第2の実施例と同様に、前記処理ステップS1412で得た結果(剰余1402)を利用する。ステップS1417でトレイラの付加と同時にトレイラ部分924~928のCRCを計算し、ステップS1318でそれらの和を取ることによって、AAL5形式のパケット1102全体のG(X)による剰余1408が計算できる。

【0206】ステップ1419でG(X)による剰余1408をG₁(X)で割った余りを求めることにより、フレームリレーのCRCが求められる。

【0207】本発明は上述した各実施例に限定されるものではなく、その要旨を逸脱しない範囲で、種々変形して実施することができる。

【0208】

【発明の効果】本発明では、プロトコル変換前後の伝送

情報中に重複部分が存在する場合、プロトコル変換の前後で行う誤り検出符号計算処理は、該重複部分に対しては変換前後において一度だけ行うので、誤り検出符号計算処理の高速化、従ってプロトコル変換処理の高速化が期待できる。

【0209】また、誤り検出符号計算処理をプロセッサ処理にて行う場合、処理負荷の大きい誤り検出符号計算処理を該重複部分に対しては一度だけで済ますことにより、処理負荷の削減を図ることができる。

【図面の簡単な説明】

【図1】本発明の一実施例に係る基本的なハードウェア構成を示す図

【図2】本実施例におけるプロセスの概念を説明するための図

【図3】本実施例のATM-FRにおけるプロセスおよび周辺ハードウェアとデータフローの関係を示す図

【図4】本実施例のFR-ATMにおけるプロセスおよび周辺ハードウェアとデータフローの関係を示す図

【図5】本実施例におけるATM(AAL5)とフレームリレーのケットフォーマットを示す図

【図6】2ポートRAMを使用したハードウェア構成の一例を示す図

【図7】イーサネットとATMのケットフォーマットを示す図

【図8】イーサネット-ATM間のプロトコル変換装置のハードウェア構成の一例を示す図

【図9】CRC計算の処理手順を示すフローチャート

【図10】CRC計算の他の処理手順を示すフローチャート

【図11】インタフェース概念図

【図12】プロトコル概念図

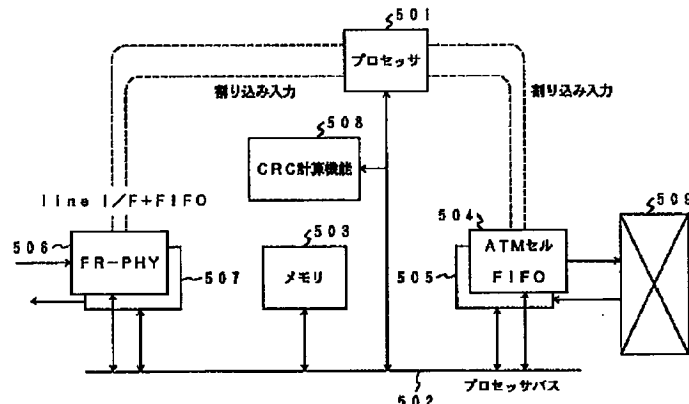
【図13】ソフト/ハードによって実現される機能分割を説明するための図

*【図14】プロトコル変換装置における処理の流れを示す図

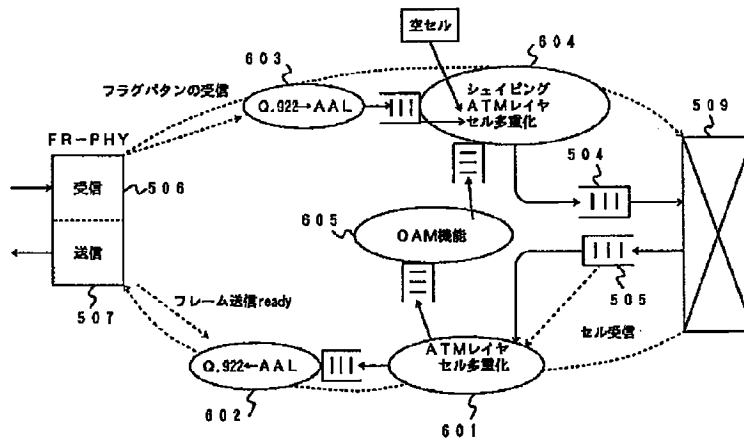
【符号の説明】

501…プロセッサ、502…プロセッサバス、503…プロセッサメモリ、504…出力セルバッファ、505…入力セルバッファ、506…フレームリレー物理レイヤ処理機能、507…フレームリレー物理レイヤ処理機能、508…CRC計算機能、509…ATMスイッチ、601…受信側ATMレイヤ機能プロセス、602…ATM-Q.922変換機能プロセス、603…Q.922-ATM変換機能プロセス、604…送信側ATMレイヤ機能プロセス、605…OAM機能プロセス、712…セルヘッダー機能変換テーブル、721…空セル処理機能サブルーチン、722…OAMセル処理機能サブルーチン、723…AAL(SAR)機能サブルーチン、732…OAMセルバッファ、741…ATM-FR変換機能プロセス、742…AALバッファ、751…バッファ、752…バッファ、753…バッファ、812…コネクション機能変換テーブル、821…OAMセル処理機能サブルーチン、822…出力セル表示、823…AAL(SAR)機能サブルーチン、824…出力セル表示、832…OAMセルバッファ、841…FR/ATM変換機能プロセス、842…AALバッファ、851…バッファ、852…バッファ、853…バッファ、1001…プロセッサ、1002…プロセッサバス、1003…受信メモリ、1004…送信メモリ、1005…出力セルバッファ、1006…入力セルバッファ、1007…受信側FR-PHY、1008…送信側FR-PHY、1010…ATMスイッチ、1201…プロセッサ、1202…プロセッサバス、1203…バッファメモリ、1204…出力セルバッファ、1205…入力セルバッファ、1206…イーサネットインタフェース、1208…ATMスイッチ

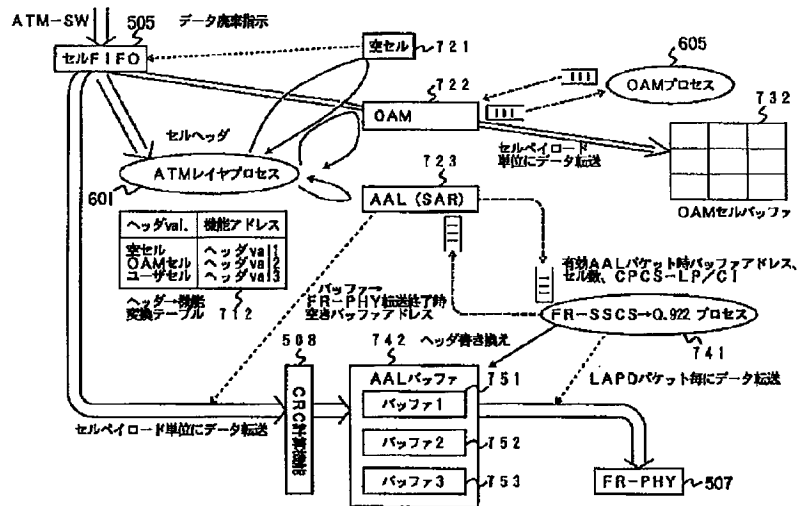
【図1】



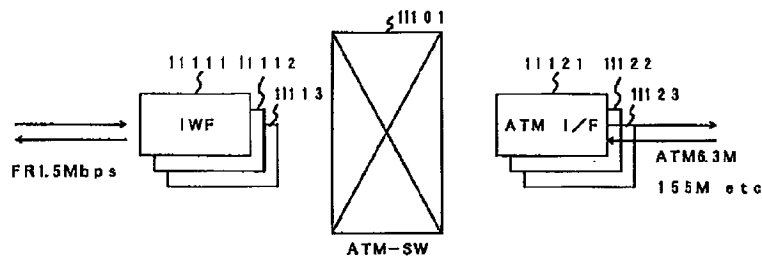
【図2】



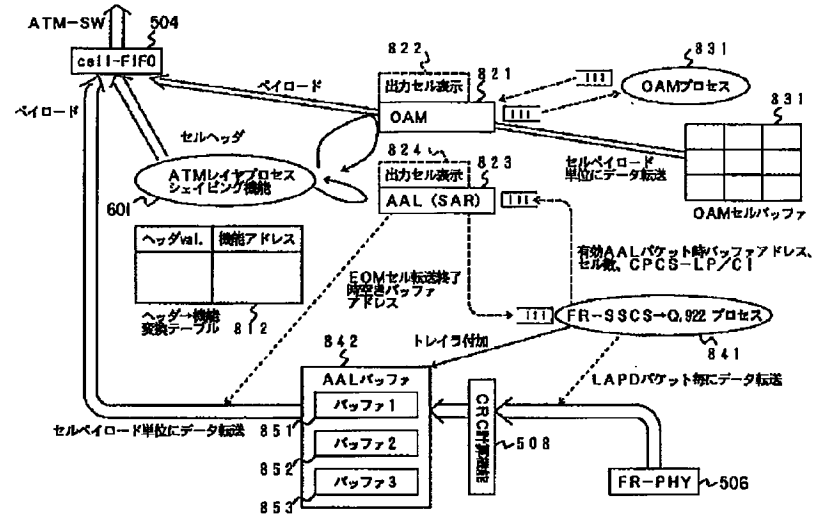
【図3】



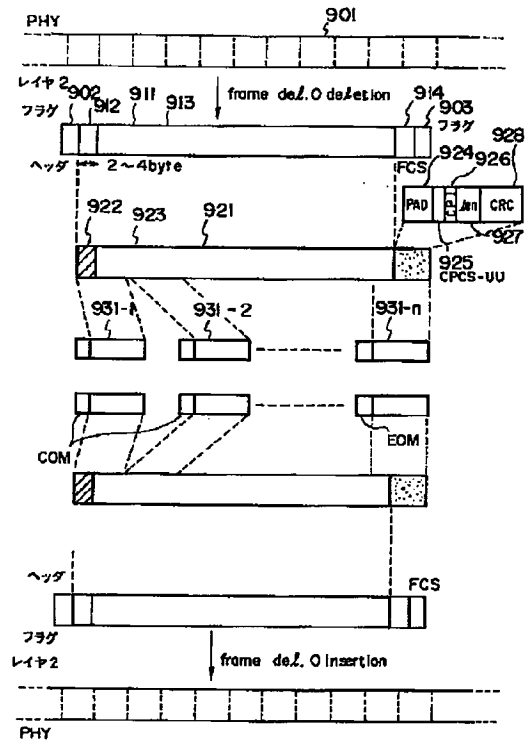
【図11】



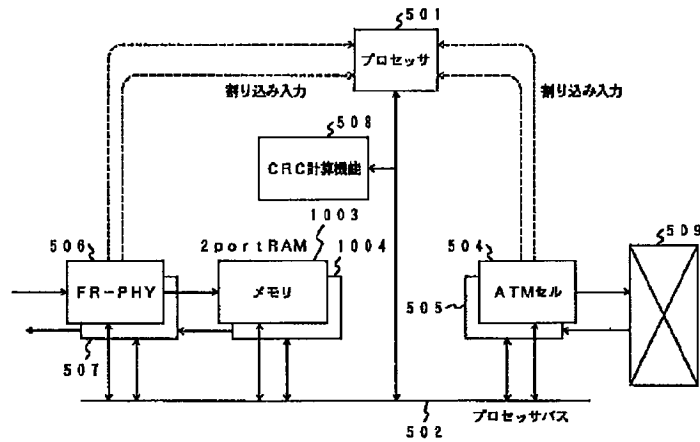
【図4】



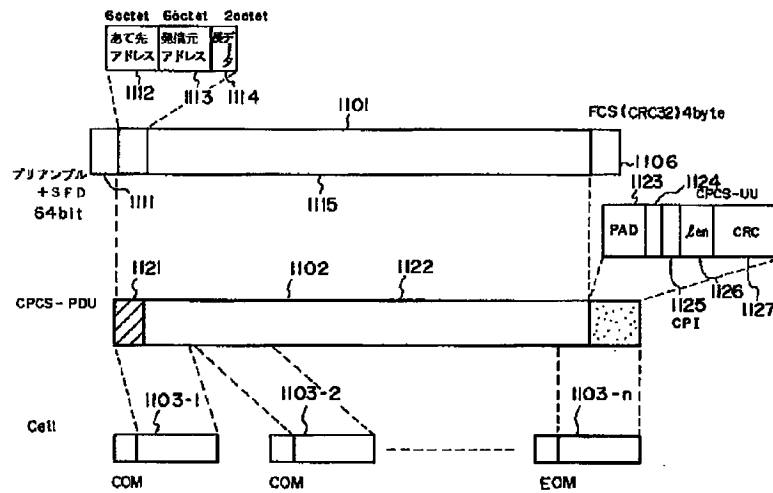
【図5】



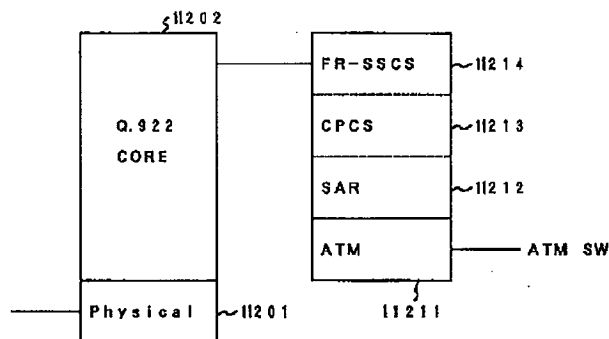
【図6】



【図7】



【図12】



1WF内部の構成

11301 プロセッサ

11302 メモリ

11303 プロセッサバス

11311 FR I/F

11321 AAL5-CPCS
ATMレイヤ機能

11322 FRパケット処理
FR-SSCS

11331 プロセッサ+メモリ+ソフトウェア

11332 プロセッサ+メモリ+ソフトウェア

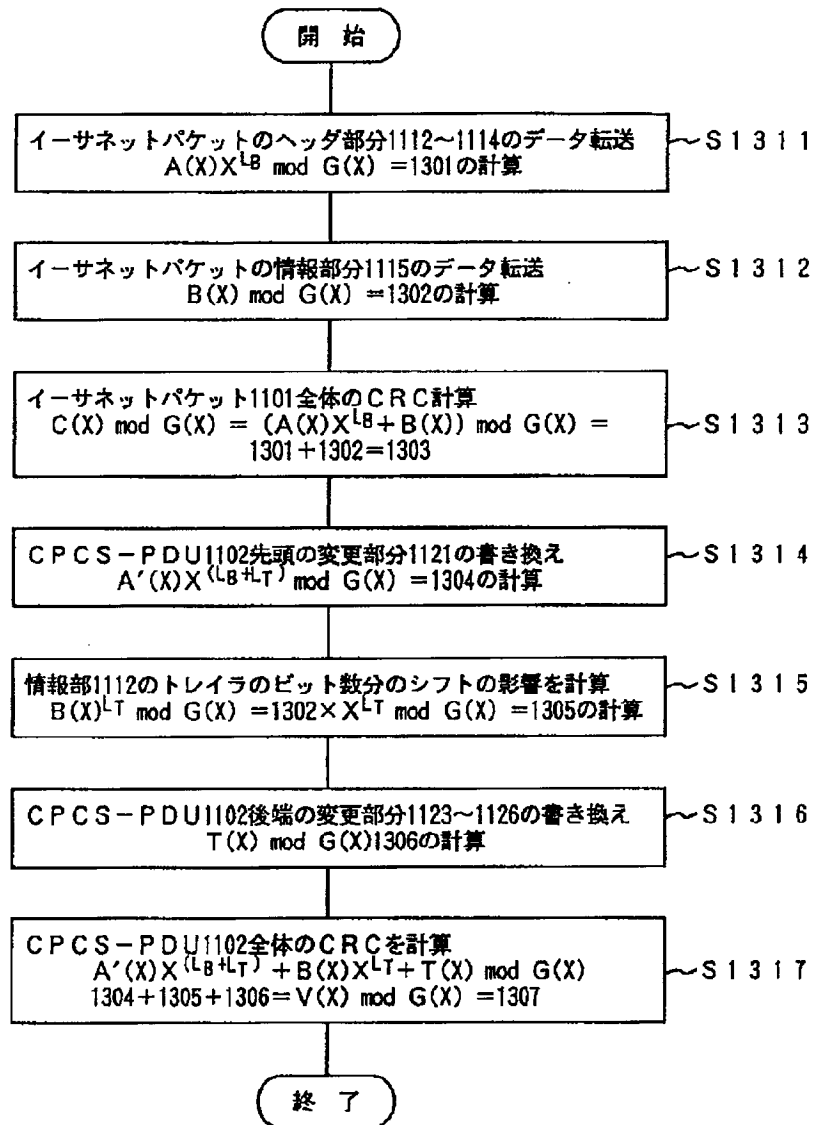
ATM I/F

SW

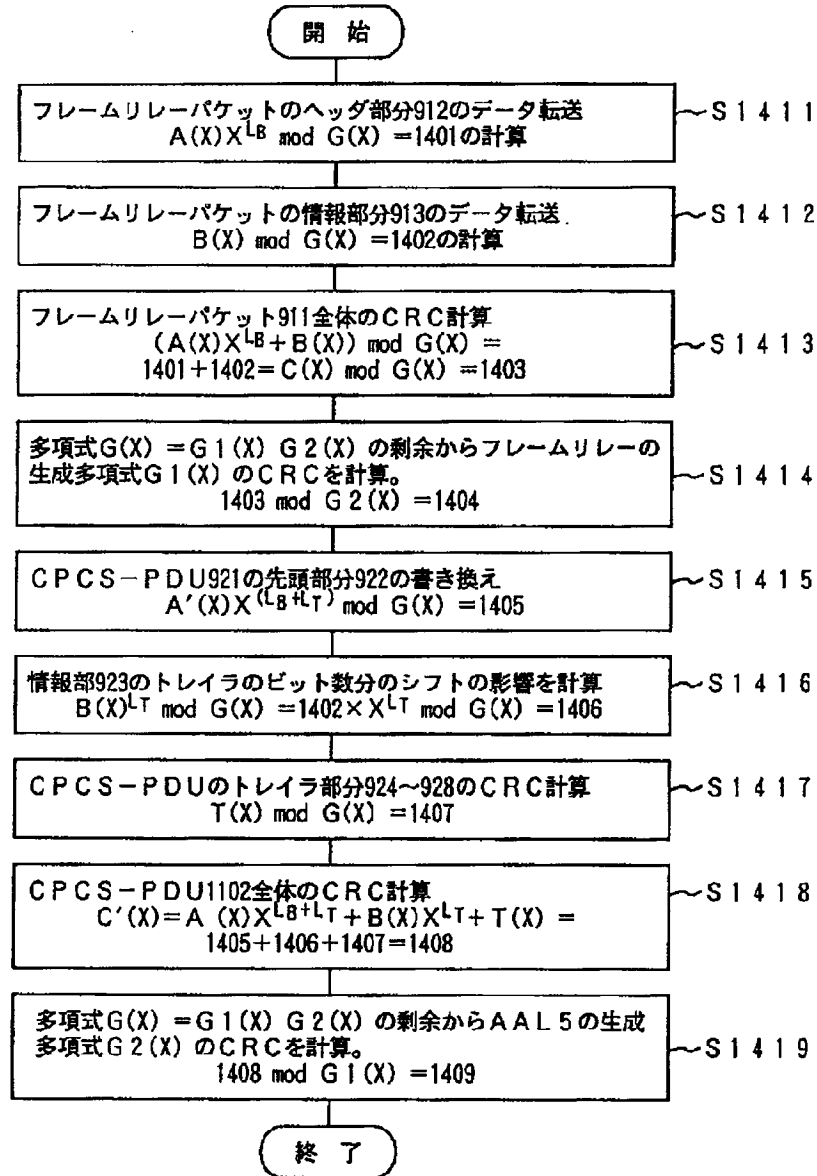
ソフトウェア処理によって実現される機能

専用ハードロジックによって実現される機能

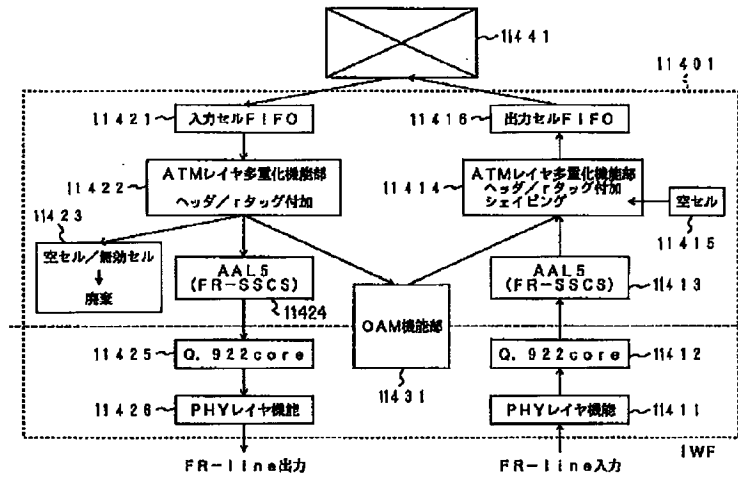
【図9】



【図10】



【図14】



フロントページの続き

(51)Int.Cl.⁶

識別記号

片内整理番号
9371-5K

F I

H 0 4 L 13/00

技術表示箇所

3 0 5 B